SECURE DATA FORWARDING
IN MULTI-HOP WIRELESS NETWORKS
FOR MOBILE USERS

Ph.D. Dissertation
of
László Dóra

Research Supervisor:
Levente Buttyán, Ph.D.

Budapest, Hungary
2011
Alulírott Dóra László kijelentem, hogy ezt a doktori értekezést magam készítettem és abban csak a megadott forrásokat használtam fel. Minden olyan részt, amelyet szó szérint, vagy azonos tartalomban, de átfogalmazva más forrásból átvettek, egyértelműen, a forrás megadásával megjelöltem.

I, the undersigned László Dóra hereby declare, that this Ph.D. dissertation was made by myself, and I only used the sources given at the end. Every part that was quoted word-for-word, or was taken over with the same content, I noted explicitly by giving the reference of the source.

A dolgozat bírálatai és a védésről készült jegyzőkönyv a Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és Informatikai Karának dékáni hivatalában elérhetőek.

The reviews of the dissertation and the report of the thesis discussion are available at the Dean’s Office of the Faculty of Electrical Engineering and Informatics of the Budapest University of Technology and Economics.

Budapest, ......................  

Dóra László
Abstract

In this thesis, I investigate security issues of two instances of multi-hop wireless networks: Wireless Mesh Networks and Delay Tolerant Networks.

A Delay Tolerant Network is a mobile ad-hoc network where the transfer of messages from their source to their destination is performed by the intermediate mobile nodes in a store-carry-and-forward manner. This means that the intermediate mobile nodes carry the messages and pass them to other intermediate nodes when they are close enough to have a connection. Since the mobile nodes are the end users, and they are who forward the messages themselves, this approach does not require the existence of any preinstalled network infrastructure. In this thesis, I give an overview of Delay Tolerant Networks and their security related problems. I address two specific issues that have to be solved in order to accomplish a reliable and secure Delay Tolerant Network: 1) stimulating cooperation, and 2) enhancing privacy in data dissemination.

Considering the first issue, a potential problem in Delay Tolerant Networks is that the quality of the services provided by the system heavily depends on the users’ willingness to cooperate. In particular, the users may act selfishly meaning that they download messages from other users that are interesting for them, but they deny storing and distributing messages on behalf of other users. Micro payment and reputation based systems can encourage the nodes to cooperate, however, all such mechanisms require centralized units in practice. In this thesis, a distributed mechanism is proposed based on the principle of barter. This mechanism assures that cooperation is the most beneficial behavior even for the selfish nodes. I build a game theoretic model to prove this statement, and the message delivery ratio and message delivery cost are investigated.

Considering the second issue, the store-carry-and-forward principle raises new aspects of the privacy problem in Delay Tolerant Networks. In particular, an attacker can build a user profile and trace the mobile nodes based on this profile even if the message exchange protocol provides anonymity. In this thesis, the attacker model is elaborated and a defense mechanism is proposed, called Hide-and-Lie. The efficiency of the attacker, the message delivery ratio and the message delivery cost are investigated when the nodes are protected with the proposed mechanism.

Wireless Mesh Networks provide last mile broadband access for mobile users who may run QoS aware applications. Wireless Mesh Networks usually consist of mesh clients, mesh routers, access points, and gateways. Gateways connect the mesh network with other networks (typically with the Internet). Access points provide access to the mesh network for the mesh clients. Finally, mesh routers route messages between two network elements within the mesh network (typically between access points and gateways). In this thesis, multi-operator maintained Wireless Mesh Networks are considered. The fact that multiple operators collaborate has many advantages such as better spectrum utilization between neighboring access points that may be operated by different operators, and extended coverage which is ensured by other operators. However, these features have special requirements which are not fulfilled in the previous proposals because they rely on inappropriate trust models. After giving an overview of the operation principles of the considered Wireless Mesh Network and its security requirements, I addressed two main issues: 1) authenticating mobile
users in the above described environment, and 2) detecting and avoiding misbehaving routers in
the network layer.

Regarding the former issue, in my research I concentrated on the authentication and access
control mechanisms that support QoS aware applications and mobility in a multi-operator envi-
ronment. Access control is essential in order to minimize the effect of injection of unauthorized
messages. After giving an overview of authentication and access control mechanism and building
up a taxonomy, I argue that none of the existing proposals can meet all the requirements that
are essential in the multi-operator maintained environment. To remedy this situation I propose
two mechanisms: 1) a scheme which is a combination of two standard methods (HOKEY and
IEEE 802.11r), and 2) a new certificate-based authentication scheme which runs locally between
the access points and the mesh clients. A novel mechanism based on weak keys is proposed for
digital signatures in order to decrease the latency of the authentication when mesh clients are not
so powerful. The proposed certificate-based authentication schemes with and without the weak
key mechanism are investigated through real implementation in a testbed environment.

Regarding the second issue, in the multi-operator environment, it is essential to detect mis-
behaving routers and to avoid them in the routing process, otherwise the quality of the services
cannot be assured. Misbehaving routers may drop data messages in order to 1) gain advantage over
competitors by dropping messages forwarded behalves of other operators, or 2) they may lie about
their metrics in order to redirect to itself as much traffic as possible, or 3) they may inject fake data
messages in order to degrade the QoS level. In order to detect misbehaving routers, a reputation
based system is proposed. Reputation values are calculated over locally maintained counters that
counts the number of forwarded messages. The proposed mechanism also takes into consideration
the fact that routers can lie fake values. The efficiency of the mechanism is investigated by means
of simulations.
A disszertációban két különböző többugrásos vezetéknelküli hálózat típus adattovábbításának biztonságát különböző módon felismerjük. Ez a két típusa a vezetéknelküli hálózatoknak a vezetéknelküli Mesh Hálózatok (Wireless Mesh Networks) és a Késleltetéstűrő Hálózatok (Delay Tolerant Networks).


Az előbbi pontot illetően, a fő probléma abban áll, hogy a Késleltetéstűrő hálózatok sokoldalúságának minősége nagyban függ a felhasználók együttműködési hajlandóságától. Egy felhasználó viselkedhet önző módon olyan értelemben, hogy kizárólag olyan üzeneteket fogad el másoktól, ami öt magát érdekli, de a többi felhasználó javára nem tárol és nem továbbít semmilyen üzenetet. Ugyan létezés mikro-fizetés vagy hírnév alapján rendszerekkel az önző csomópontokat is együttműködésre lehet bírni, de a gyakorlatban ezen rendszerek működtetéséhez központi elenhelyeből van szükség, ami sok esetben távol áll a Késleltetéstűrő hálózatok alapelvétől. A disszertációban egy barter alapú elosztott eljárás kerül bemutatásra. Ez az eljárás biztosítja, hogy az együttműködés akkor is a legkifizetődőbb viselkedésmod legyen, ha a csomópontok önzőek. Ezen állítást játékelméleti modellben bizonyítom, és megvizsgálok az üzenetek későbbesítési arányát, valamint a barter alapú eljárás többlet költségét.

A második pontot illetően, a Késleltetéstűrő hálózatokban a store-carry-and-forward elv új fajta problémát vet fel a privacy védelmével szemben. Amennyiben egy támadó fel tud építeni felhasználói profilokat a tárolt üzenetek alapján, akkor is képes lehet a felhasználókat követni, ha a kommunikáció egyébként anonim. A disszertációban a támadó model kidolgozása után egy ún. Hide-and-Lie védelmi eljárást javaslok. A támadó sikereségét, az üzenetek későbbesítési arányát és a védelmi mechanizmus többlet költségét is vizsgálok.

Vezetéknelküli Mesh Hálózatok (Wireless Mesh Networks) legfőbb célja szélesásvú hozzáférés biztosítása mobil felhasználók számára, akik QoS érzékeny alkalmazást futtathatnak eszközökeiken. Többnyire mesh klienkek, access pointok, mesh routerok és gatewayek alkotják a Vezetéknelküli Mesh Hálózatokat. A gatewayek (átjáró) biztosítják a kapcsolatot a mesh hálózat és más típusú hálózatok (tipikusan Internet) között. Az access pointok biztosítanak a mesh hálózathoz hozzáférést a mesh klienkek számára. Végül a mesh routerok továbbítják az üzeneteket két hálózati elem között a mesh hálózaton belül (tipikusan egy access point és egy gateway között). A disszertációban a több operátor által üzemeltetett Vezetéknelküli Mesh Hálózatokkal foglalkozok, melyek előnye a jobb spektrum kihasználás a szomszédos, de idegen access pointok között és a nagyobb lefedettség,
amit az idegen access pointok használata biztosít a többi operátor számára. Ugyanakkor a speciális körülmények speciális követelmények kielégítését igénylik a tervezőktől, melyeket a jelenleg ismert megoldások csak részben tudnak teljesíteni, mivel más bizalmi modellben gondolkodtak a kifejlesztők. A Vezetéknelküli Mesh Hálózatok működési elveinek és biztonsági követelményeinek áttekintése után a következő két feladatot azonosítottam: 1) mobil felhasználók hitelesítése a fent leírt környezetben, és 2) rosszul viselkedő mesh routerek azonosítása és kikerülése hálózati rétegeben.

Az előbbi feladattal kapcsolatban kutatásaim során olyan hitelesítő és hozzáférés-védelmet biztosító mechanizmussal foglalkoztam, mely támogatja, hogy a mobil felhasználó QoS érzékeny alkalmazásokat is futtassanak. A hozzáférés védelem biztosítása rendkívüli fontos a jogosulatlan üzenetek injektálásának megakadályozása miatt. A létező megoldások áttekintése és kategorizálása után megállapítottam, hogy egyik jelenlegi megoldás sem elégté ki azon követelményeket, melyek a több operátor általi üzemeltetés támasza a hitelesítő mechanizmussal szemben. Két megoldást is javasoltunk: 1) egy olyan sémát, amely két standard megoldás kombinációja (HOKEY és IEEE 802.11r), és 2) egy új lokális tanúsítvány alapú hitelesítő eljárást, amely csak az access point és a mesh kliens között fut le. A hitelesítés felgyorsítása érdekében egy ún. gyenge kulcs alapú eljárást is javaslunk a digitális aláírásban kevésbé erős teljesítményű klienek számára. A javasolt tanúsítvány alapú hitelesítési sémá teljesítményét egy valós implementáció keresztül vizsgálom a gyenge kulcsú eljárással és anélkül.

A második feladatot illetően, a több operátor által üzemeltetett környezetben lényeges, hogy a elvárt viselkedéstől eltérő routereket azonosítsuk és a további útvonalválasztásból kizárjuk, különben a hálózat által igért QoS nem biztosított. A helytelenül viselkedő routerek 1) hogy előnyt szereznek a közvetlen konkurenselként szemben, a más operátor ügyfeleire továbbított üzeneteket elődöhatják, 2) hazudhatnak az öket érintő metrikákkal kapcsolatban, hogy minél több forgalom haladjon rajtuk keresztül, vagy 3) hamis üzeneteket injektálhatnak a hálózatba, hogy a QoS szintet csökkentsék. A helytelenül viselkedő routerek felfedezéséhez egy hírnév alapú rendszert fejleszttettem ki. A hírnevet meghatározó érték kiszámításához mindeniket lokálisan jegyzem, hogy mennyi üzenetet továbbított egy adott útvonalon, mely adatokat felhasználva a gatewayek megbizhatósági értéket számolnak a mesh routerekhez. A javasolt megoldás figyelembe veszi, hogy a rosszul viselkedő routerek hamis értékeket küldhetnek. A javasolt megoldás hatékonysségi szimuláció keresztül mutatom be.
Acknowledgement

First of all, I would like to express my deep and sincere gratitude to my supervisor, Professor Levente Buttyán, Ph.D., Department of Telecommunication, Budapest University of Technology and Economics. His wide knowledge, his logical way of thinking and overwhelming work spirit have been of great value for me. His understanding, encouraging and personal guidance have provided a good basis for the present thesis.

I am also grateful to the current and former members of the CrySyS Laboratory: Boldizsár Bencsáth, László Czap, László Csić, Amit Dvér, Tamás Holczer, Péter Schaffer, Vinh Thong Ta, and István Vajda for the illuminating discussions on different technical problems that I encountered during my research. They also provided a pleasant atmosphere which was a pleasure to work in.

I would also like to thank for our joint efforts and publications to Ioannis Askoxylakis, Gergely Ács, Boldizsár Bencsáth, András Bohák, Márk Félegyházi, Tamás Holczer, Fabio Martinelli, Marinella Petrocchi, Vasílios Siris, Dávid Szili, and István Vajda.

I am thankful to the reviewers of my thesis, Sonja Buchegger and Tibor Remsző. I am grateful that they reviewed and evaluated my thesis in spite of the limited time that they had.

The financial support of the Mobile Innovation Centre (MIK) and the support of the BIONETS (FP6-027748) and EU-MESH (FP7-215320) EU projects are gratefully acknowledged. I am also grateful to the members of the EU projects for the heartening atmosphere during the meetings.

I am indebted to my family: my mother Mária, my father István, my godfather József, my grandmothers Rozália, Magdolna, my aunt Erzsébet, and her husband István for their support and encouragement during my whole life. I am especially grateful to my mother encouraging me to undertake this PhD programme. I am also grateful to my mother-in-law Tünde, and my father-in-law Bélá for accepting me in their family.

I express hereby my thanks to my wife, Adrienn. She guaranteed me such a stable background that facilitated for me to focus on my research. Without her help and constant support, it would have been impossible to finish this dissertation. It meant a lot to me that she always stands by me. Last but not least, I would like to thank the patience of my daughters, Kincső and Villő. I hope that they will excuse that sometimes I had to work instead of taking care of them.
# Contents

1 Introduction

1.1 Delay Tolerant Networks .................................................. 1
   1.1.1 Description of technology ........................................ 1
   1.1.2 Security issues ..................................................... 3
   1.1.3 Research objectives ............................................... 5
   1.1.4 My achievements ................................................ 5

1.2 Wireless Mesh Networks .................................................. 6
   1.2.1 Description of technology ........................................ 6
   1.2.2 Security issues ..................................................... 8
   1.2.3 Research objectives ............................................... 10
   1.2.4 My achievements ................................................ 10

2 Stimulating cooperation using barter in Delay Tolerant Networks 13

2.1 Introduction ........................................................................ 13

2.2 State-of-the-art ............................................................... 14

2.3 System analysis ............................................................... 14
   2.3.1 System model ......................................................... 15
   2.3.2 Convergence of the goodput ........................................ 17
   2.3.3 Simulations ............................................................ 19
   2.3.4 Conclusion ............................................................ 22

2.4 My approach ...................................................................... 23

2.5 Game model ....................................................................... 25

2.6 Results .............................................................................. 26

2.7 Future work ........................................................................ 29

2.8 Summary ............................................................................ 31

3 Hide-and-Lie for enhancing privacy in Delay Tolerant Networks 33

3.1 Introduction ........................................................................ 33

3.2 State-of-the-art ................................................................... 33

3.3 System model ..................................................................... 34

3.4 Attacker model ................................................................... 36
   3.4.1 Leaking information .................................................. 36
   3.4.2 Attacker behavior ...................................................... 38
   3.4.3 Attacker functions .................................................... 38

3.5 My approach ...................................................................... 39

3.6 Simulations ........................................................................ 40

3.7 Results ............................................................................... 42

3.8 Future work ........................................................................ 46

3.9 Summary ............................................................................ 47
CONTENTS

4 Fast authentication methods in Wireless Mesh Networks 49
  4.1 Requirements on authentication at the link layer 49
  4.2 State-of-the-art and design options 50
    4.2.1 Taxonomy 50
    4.2.2 Existing proposals 51
    4.2.3 Summary 56
  4.3 Selected approaches and their rationale 57
  4.4 IEEE 802.11r and HOKEY in a multi-operator environment 58
    4.4.1 Architecture 58
    4.4.2 Initial authentication 59
    4.4.3 Inter-domain handover 60
    4.4.4 Intra-domain handover 60
    4.4.5 Fulfillment of the requirements 60
    4.4.6 Implementation issues 61
  4.5 Certificate based authentication and access control 61
    4.5.1 Architecture 62
    4.5.2 Design rationale 62
    4.5.3 Certification based authentication and key transport protocols 63
    4.5.4 Public key algorithms and key parameters 65
    4.5.5 Cross-certificates 68
    4.5.6 Performance analysis 69
    4.5.7 Evaluation 74
  4.6 Future work 75
  4.7 Summary 75

5 Misbehaving router detection for Wireless Mesh Networks 77
  5.1 Introduction 77
  5.2 Security of routing protocols 77
  5.3 State-of-the-art 78
  5.4 My approach 80
  5.5 System and attacker model 81
    5.5.1 System model 81
    5.5.2 Attacker model 81
  5.6 Node Trust Value 82
    5.6.1 Calculation of Node Trust Value on each route 82
    5.6.2 Aggregation of Node Trust Values 83
    5.6.3 Utilizing the Node Trust Value aggregated by the access points 84
  5.7 Performance analysis 84
    5.7.1 Simulations 84
    5.7.2 Results 85
  5.8 Future work 89
  5.9 Summary 89

6 Application of new results 91

7 Conclusion 95

Appendix A Authentication related standards 99
  A.1 IEEE 802.11i 99
  A.2 IEEE 802.11r 102
    A.2.1 General description 102
    A.2.2 Mobility domains 102
    A.2.3 Initial authentication 103
    A.2.4 Key hierarchy 104
 |
|-----------------|----------------|---|
| **A.2.5** Re-authentication within a mobility domain | 104 |
| **A.3** HOKEY | 105 |
| **A.3.1** General description | 106 |
| **A.3.2** Key hierarchy | 106 |
| **A.3.3** Initial authentication | 107 |
| **A.3.4** Re-authentication | 108 |
| **A.3.5** Key delivery | 108 |

**Appendix B** Time consumption of asymmetric cryptographic primitives 111
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Illustration of the Wireless Mesh Networks</td>
<td>6</td>
</tr>
<tr>
<td>2.1</td>
<td>Devaluation of primary messages over time</td>
<td>16</td>
</tr>
<tr>
<td>2.2</td>
<td>The convergence of the goodput of some sample nodes</td>
<td>17</td>
</tr>
<tr>
<td>2.3</td>
<td>The convergence of the average goodput and its empirical deviation</td>
<td>18</td>
</tr>
<tr>
<td>2.4</td>
<td>Simplified map of Budapest used in SUMO mobility model</td>
<td>20</td>
</tr>
<tr>
<td>2.5</td>
<td>Histogram of the duration of getting from a meeting point to another neighboring one</td>
<td>21</td>
</tr>
<tr>
<td>2.6</td>
<td>Goodput without incentives</td>
<td>22</td>
</tr>
<tr>
<td>2.7</td>
<td>Best responses</td>
<td>27</td>
</tr>
<tr>
<td>2.8</td>
<td>Gain of player NULL</td>
<td>28</td>
</tr>
<tr>
<td>2.9</td>
<td>Message download statistics of RRW</td>
<td>29</td>
</tr>
<tr>
<td>2.10</td>
<td>Histogram of Nash Equilibrium values</td>
<td>30</td>
</tr>
<tr>
<td>2.11</td>
<td>Goodput with barter mechanism</td>
<td>30</td>
</tr>
<tr>
<td>3.1</td>
<td>Message exchange protocols</td>
<td>37</td>
</tr>
<tr>
<td>3.2</td>
<td>Success probability of $\mathcal{A}$ as a function of the Hide-and-Lie strategy values</td>
<td>43</td>
</tr>
<tr>
<td>3.3</td>
<td>Analytically determined upper bound for success probability of ideal $IP$ based attacker functions</td>
<td>45</td>
</tr>
<tr>
<td>3.4</td>
<td>Average gain with the empirical standard deviation</td>
<td>45</td>
</tr>
<tr>
<td>3.5</td>
<td>Average costs</td>
<td>46</td>
</tr>
<tr>
<td>4.1</td>
<td>Architecture of IEEE 802.11r authentication method in multi-operator environment</td>
<td>59</td>
</tr>
<tr>
<td>4.2</td>
<td>Nonce based authentication</td>
<td>64</td>
</tr>
<tr>
<td>4.3</td>
<td>Timestamp based authentication</td>
<td>65</td>
</tr>
<tr>
<td>4.4</td>
<td>Nonce based authentication with weak key mechanism</td>
<td>68</td>
</tr>
<tr>
<td>4.5</td>
<td>Timestamp based authentication with weak key mechanism</td>
<td>69</td>
</tr>
<tr>
<td>4.6</td>
<td>Authentication protocols embedded in EAP framework</td>
<td>70</td>
</tr>
<tr>
<td>4.7</td>
<td>Average authentication delay</td>
<td>72</td>
</tr>
<tr>
<td>4.8</td>
<td>Message by message comparison of authentication delay with or without weak key mechanism</td>
<td>73</td>
</tr>
<tr>
<td>4.9</td>
<td>Message by message comparison of authentication delay with or without cross certificates</td>
<td>74</td>
</tr>
<tr>
<td>5.1</td>
<td>Sample scenario</td>
<td>86</td>
</tr>
<tr>
<td>5.2</td>
<td>Average Node Trust Value grouped by different node categories</td>
<td>86</td>
</tr>
<tr>
<td>5.3</td>
<td>Average numbers of dropped data packets</td>
<td>87</td>
</tr>
<tr>
<td>5.4</td>
<td>Average lengths of the routes</td>
<td>87</td>
</tr>
<tr>
<td>5.5</td>
<td>Node Trust Value adaptation</td>
<td>88</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>A.1</td>
<td>IEEE 802.1X authentication</td>
<td>100</td>
</tr>
<tr>
<td>A.2</td>
<td>IEEE 802.11i four-way handshake</td>
<td>101</td>
</tr>
<tr>
<td>A.3</td>
<td>Initial IEEE 802.11i authentication in multi-operator environment</td>
<td>102</td>
</tr>
<tr>
<td>A.4</td>
<td>Initial authentication in IEEE 802.11r</td>
<td>103</td>
</tr>
<tr>
<td>A.5</td>
<td>Reassociation in IEEE 802.11r (Fast Transition Protocol)</td>
<td>105</td>
</tr>
<tr>
<td>A.6</td>
<td>Key hierarchy in HOKEY</td>
<td>106</td>
</tr>
<tr>
<td>A.7</td>
<td>HOKEY initial authentication in a foreign network</td>
<td>108</td>
</tr>
<tr>
<td>A.8</td>
<td>HOKEY re-authentication in a foreign network</td>
<td>109</td>
</tr>
<tr>
<td>B.1</td>
<td>Time consumption of key exchange algorithms</td>
<td>113</td>
</tr>
<tr>
<td>B.2</td>
<td>Time consumption of digital signature algorithms</td>
<td>114</td>
</tr>
<tr>
<td>B.3</td>
<td>Time consumption of encryption algorithms</td>
<td>114</td>
</tr>
</tbody>
</table>
# List of Tables

2.1 Parameter values of the simulations .......................... 21  
3.1 Fixed simulation parameters ..................................... 41  
3.2 Varied simulation parameters .................................. 41  
3.3 Parameter values of investigated scenarios .................... 42  
4.1 Categorized list of the proposed authentication methods .......... 52  
4.2 Requirements and proposed solution for connection key generation .... 56  
4.3 Requirements and authentication methods .......................... 57  
5.1 Fixed parameter values of the simulations ....................... 85  
5.2 Varying parameter values of the simulations ..................... 85  
B.1 Test elliptic curve parameters .................................. 112
Chapter 1

Introduction

In this thesis, I investigate security issues of two instances of multi-hop wireless networks: Delay Tolerant Networks and Wireless Mesh Networks. In this chapter, the two technologies are described separately based on [Pelusi et al., 2006; Akyildiz et al., 2005]. I set the focus on the scenarios I investigate, and I give an overview of security issues ([Farrell et al., 2009; Askoxylakis et al., 2009]) that can be addressed in the considered scenarios. Finally, the research objectives and a short overview of my results are given after the overviews.

1.1 Delay Tolerant Networks

1.1.1 Description of technology

A Delay Tolerant Network (DTN) is an infrastructureless network, where the message dissemination is performed by the participating mobile — usually battery driven — end-nodes. The messages are delivered in a store-carry-and-forward manner. With this approach, the messages can be delivered even if an online end-to-end route connecting the source and the destination never exists. This means that the intermediate mobile nodes carry the messages and pass them on to other intermediate nodes when they have a connection (e.g., when they are in vicinity).

The terms Delay Tolerant Network and Opportunistic Network are usually used interchangeably in the literature. However, some researchers assign wider interpretation to Opportunistic Networks, where the opportunistic attribution describes the routing mechanism independently of the network architecture, either infrastructure based, or infrastructureless. In this thesis, the two terms are used interchangeably.

DTN can be considered as a branch of MANET (Mobile Ad hoc NETwork). Since in this thesis, only DTN specific issues are considered and the proposed solutions are specific to the DTN, I do not give a detailed description of the MANET, but DTN.

Only special applications can take advantage of DTNs that promises low cost maintenance due to the infrastructureless approach. The applications must tolerate the delay caused by the lack of online routes, however, there are wide range of applications.

Wildlife monitoring [Juang et al., 2002] and Internet service providing in rural areas [Pentland et al., 2004] are those applications which have been already implemented. In this thesis, I address the issues of delay tolerant personal wireless networks. These networks typically consist of handheld devices owned by mobile users and local information needs to be distributed to a set of nearby destinations based on their interest in the information.

As an example, let us consider a touristic city, such as Rome or Paris, where it would be beneficial for the tourists to be able to share information concerning the various touristic sights. A possible solution would be to set up an on-line bulletin board where tourists can post messages of potential interest for other tourists. However, this solution needs a service provider that runs
the bulletin board service, and each tourist must have wireless Internet access for posting and downloading messages. The business model behind this solution would likely require the tourists to pay for both the service usage and the network access.

An alternative solution could benefit from the proliferation of Bluetooth capable personal devices such as mobile phones, PDAs, and MP3 players. These devices can communicate with each other when they are in vicinity even without any user intervention. Touristic information can then be distributed in a store-carry-and-forward manner by using these devices and by exploiting the mobility of the tourists themselves. This would result in a city-wide Delay Tolerant Network.

There are different sub-instances of the DTNs. They can be differentiated by the way the messages are addressed, by the method of data delivery, and by the existence or non-existence of backhaul infrastructure.

**Addressing methods.** Destinations can be addressed either in a target centric manner, or in a data centric manner. In the target centric manner, the recipient of the data is known, and the task is to deliver the data to that user. In the data centric manner, only the data is known, and the recipient can be anyone who is interested in that particular data. The task here is to deliver the information to as many interested users as possible.

**Data delivery methods.** Data delivery methods can be differentiated by the number of replicas of messages. In the case of 1-copy approach, intermediate nodes passing the message towards the destination delete the message from the memory immediately. In the case of replication based approach, the intermediate nodes are allowed to hold the message after passing the message to other intermediate nodes, too.

Replica based data forwarding mechanism can be dissemination based or context based. Different data forwarding mechanisms can be developed depending on how the destination is determined. The dissemination based approach suits better data centric addressing, and context based algorithms suit better target centric addressing.

At dissemination based algorithms there is no a priori knowledge of possible routes towards the destination or destinations. Because of that and the fact that the destinations are not known either, each message must be disseminated all over the network. The basis of dissemination based algorithms is flooding and they differ on how they limit the number of message copies. This approach is usually resource hungry because the nodes have to store and forward many messages. Dissemination based approaches are proposed in e.g., [Vahdat and Becker, 2000; Lindgren et al., 2004; Spyropoulos et al., 2005].

The context based algorithms require some knowledge about the network topology. The nodes have to maintain information of the other participating nodes. The best relay node towards the destination is selected based on this information. This approach reduces the message duplicates, but on the other side, increases the delay because of the unexploited and unpredictable opportunities. There are many context based algorithms proposed in the literature (see e.g., [Musolesi et al., 2005; Leguay et al., 2005]).

**Base station.** DTNs can work free of preinstalled infrastructure as the touristic application showed or they can be an extension of other infrastructure based networks which require the existence of base stations that serve as gateways between the two network (e.g., providing Internet in rural areas). If base stations exist, usually one end point of the communication is a base station. The base station can be fixed or mobile.

Since in the considered applications, the destinations are defined by their interests (which may include location information), my approach relies on data centric addressing method. Due to the data centric method, the data packets are forwarded based on a dissemination approach. I aim to propose solutions that do not require any infrastructure element.
1.1.2 Security issues

Attacker model

In order to be able to efficiently protect the data forwarding mechanism in the DTNs, the potential attacker must be identified. Therefore, first, I classify the attackers and describe their objective and their tools.

Classes of attackers The following three types of attackers can be differentiated regarding to the data dissemination process:

- **Selfish node:** An internal node which participates in the data dissemination process and it is also potentially a source or a destination of a message. A selfish node performs an attack only if it has a direct gain by doing so.

- **Malicious node:** Similarly to the selfish node, it is a participant of the network, however, its gain is realized out of the network which is not investigated. A malicious node can be viewed as someone who simply cause damage to the network.

- **External attacker:** Similar to the malicious node, but it has only limited access to the network.

Objectives of attacks. The main objectives of attacks can be the followings:

- **Denial-of-Service (DoS):** A malicious node or an external attacker may want to degrade the performance of the data dissemination.

- **Violate users’ privacy:** A malicious node or an external attacker may want to violate users’ privacy obtaining their interest profile or their movement trace.

- **Free-ride:** Selfish users may want to exploit the network receiving or sending messages, but they deny to forward messages on behalf of other nodes.

- **Flood the network with SPAM:** Malicious nodes may flood the network with SPAMs exploiting that the honest nodes forward the message even if they are neither the source, nor the destination of the message.

- **Force other nodes prioritizing the attacker’s message:** Selfish nodes may try to force the intermediate nodes to prioritize its messages in order to increase the delivery ratio or to decrease the delay of arrival of their messages. This may result in producing more replicas of the attacker’s message in the network.

Attack mechanisms. An attacker can use the following two techniques to apply the above described attacks:

- Attacks on wireless communications including eavesdropping, jamming, replay, and injection of messages, and traffic analysis.

- Attacks on the data dissemination mechanism including dropping messages or altering some parameters of messages such that it changes the behavior of other participants.

Security requirements

Stimulating cooperation. Note that due to the highly distributed manner of DTNs, in the case of most of data forwarding mechanisms, a DoS attack performed by a single external attacker has only limited effect in respect of time and space. However, large scale selfishness can lead to a DoS attack that has to be handled. A potential problem in DTNs is that the quality of the service provided by the system heavily depends on the users’ willingness to cooperate. In particular, the users may act selfishly meaning that they download messages from other users that are interesting
for them, but they deny storing and distributing messages for the benefit of other users. As shown in [Panagakis et al., 2007], if the majority of the users behave selfishly, then the message delivery rate decreases considerably and the quality of the services provided by the network decreases accordingly. A new mechanism has to be developed which assures that it is worth to cooperate either for selfish nodes.

**Preventing SPAM.** The fact that end-users forward messages on behalf of other nodes too can be exploited by spammers. An adversary may inject SPAM messages into the system which slow down the dissemination of valid messages, however some anti-spam techniques based on content analyzing used in the Internet can be applied by the nodes to prevent the spreading of SPAM messages and to limit the effect of unwanted traffic.

Furthermore, in DTNs they can be more effective than in the Internet, because they are applied in the end systems, and therefore they do not prevent the increased usage of the bandwidths. In contrast to this, in DTNs, spam filtering can be implemented by the nodes which are not only end systems but forwarding nodes. Thus, unwanted traffic can be stopped immediately as it enters in the network, and it does not harm the entire system.

Considering some anti-spam techniques [Goodman et al., 2007], the bayesian approach [Sahami et al., 1998], statistical compression models [Bratko et al., 2006], and using regular expressions are the techniques that suit DTNs.

**Fairness.** Selfish nodes should be unable to force other nodes to prioritize their message. Otherwise, the attacker’s message could be disseminated in the network faster than the others. This attack can be performed by changing some parameters related to the messages such as e.g., date of the origin. However, it depends on the particular data dissemination protocol which parameter needs to be changed for the successful attack.

A particular data dissemination protocol, called binary spray-and-wait algorithm [Spyropoulos et al., 2005], can be attacked as follows. The source of a message sets a counter bounded to the message. This counter determines the number of copies that should spread in the network. A forwarder node passes half of the copies to each encountered node. Each nodes maintain how many copy of the messages they store. When only one copy remains at a forwarder node, it passes that message to those nodes only that are interested in the message. In the original protocol nothing prevents selfish nodes from increasing their counter. The number of copies can be increased continuously in the network over a limit defined by the protocol. This attack may decrease the bandwidth.

The sensitive parameters of the data dissemination protocol must be protected against such attacks. Whether the parameter can be protected depends on the particular protocol.

**Preserving privacy.** Without privacy protection, no new technology should spread widely. The privacy of the users must be ensured in DTNs as well. Some of the problems can be mitigated by traditional technologies, but new problems are introduced by the store-carry-and-forward manner of the DTNs such as the one described below.

It is essential that the communication be anonymous. Anonymity (or at least pseudonymity) can be easily achieved by the usage of pseudonyms (i.e., temporal identifiers) [Pfitzmann and Kühntopp, 2001]. A more serious and DTN specific privacy problem is that the nodes can be identified by their stored messages. If an attacker is able to build an interest profile of a user using the exchanged application data, the user becomes traceable even if the communications are completely anonymous. Therefore, a new mechanism or an adaptation of some proposed mechanisms is required in DTNs to ensure untraceability of the nodes, namely, to prevent an attacker to build traceable user profiles.
1. INTRODUCTION

1.1.3 Research objectives

So far, four main issues have been addressed in delay tolerant personal wireless networks: 1) stimulating cooperation, 2) preventing SPAM, 3) providing fairness, and 4) preserving privacy. I showed that the problem of SPAM can be solved with existing solutions. The investigation of fairness seems to be protocol-specific problem. In this thesis, I consider the DTNs in a more abstracted way. Therefore, in this thesis, these problems are not considered. The problems I consider are 1) related to stimulating the cooperation in data dissemination, and 2) related to the privacy of the users, particularly, to their traceability.

As I have already described, it is essential to prevent selfish behavior in data dissemination, because the data forwarding in DTN relies on the end-users’ willingness to help each other. Current centralized solutions, such as some reputation or micropayment schemes, do not suit well the DTN environment, because they require a trusted third party. The distributed reputation mechanisms can hardly manage the huge number of participants or their trust model are inappropriate in many scenarios. My main goal is to propose a distributed mechanism that encourages the nodes to store, carry, and forward messages even if they are not particularly interested in their contents. The mechanism should decrease the delivery delay and increase the delivery ratio.

I addressed the problem of traceability of users participating in DTNs. The traceability of the users can be a problem on any layer of the communication stack. In this thesis, I investigate the layer where the store-carry-and-forward raise a DTN specific problem. In particular, an attacker can build a user profile of a node based on what messages the node stores and what messages it wants to download. After profiling, the attacker can trace the node even if the node communicates with the other nodes through anonymous links. As far as I know, I am the first, who come up with this issue. I aim to propose a defense method against the above described attackers without jeopardizing the node’s main goal, the message collection.

1.1.4 My achievements

In Chapter 2, I show the approach I propose to stimulate the cooperation of nodes which is based on the principles of barter. More specifically, I require that when two nearby nodes establish a connection, they first send the description of the messages that they currently store to each other, and then they agree on which subset of the messages they want to download from each other. In order to ensure fairness, the selected subsets must have the same size, and the messages are exchanged in a message-by-message manner, in preference order.

Note that it is entirely up to the nodes to decide which messages they want to download from each other. They may behave selfishly by downloading only those messages that are of primary interest for them. However, selfish behavior may not be beneficial in the long run. In particular, the idea is that a message that is not interesting for a node A may be interesting for another node B, and A may use it to obtain a message from B that is indeed interesting for A. In other words, the messages that are not interesting for a node still represent a barter value for the node, and hence, it may be worth downloading and carrying them. Hereby, the messages can be viewed as an investment to get new interesting messages later.

I introduce my proposed mechanism as a symmetric non-cooperative game to analyze the behavior of the nodes using game theory. I show that the barter-based approach indeed discourages selfishness. More precisely, the analysis shows that it is worth for users collecting, carrying and disseminating messages even if they are not interested in them, which has a positive effect on quality of data dissemination. In particular, the results show that, in realistic scenarios, the message delivery rate considerably increases if the nodes follow the Nash Equilibrium strategy in the barter mechanism compared to the data dissemination protocol when no encouraging mechanism is present.

In Chapter 3, I investigate the traceability of users. For this, I built a system and an attacker model, and I proposed some attacker functions.

In order to enhance users’ privacy, two simple methods can be used to modify the user profile that an attacker can learn at a single time. The first one is to hide some interesting categories,
1. INTRODUCTION

and claim them as uninteresting. The second one is to lie about some uninteresting categories, and
claim them as interesting. These techniques can be used at the same time, this is why I call this
strategy Hide-and-Lie. I give a simple but rather general solution: every node generates its current
user profile from its original user profile by indicating an uninteresting category as interesting or
vice versa with a given probability. With tuning this probability, the nodes can select between
high privacy level and low data-forwarding overload.

In my model, I analyzed the efficiency of both the attacks at different parameter values and the
proposed defense mechanism. I showed that the success probability of an attacker can be decreased
substantially with the proposed Hide-and-Lie Strategy. The message delivery ratio and the costs
at different Hide-and-Lie parameter values are also investigated.

1.2 Wireless Mesh Networks

Similarly to the previous section, firstly, I describe the technology itself. Then, I give an overview
of the security issues arisen in the considered scenarios. Finally, the research objectives and a short
overview of my results are given.

1.2.1 Description of technology

As shown in Figure 1.1, a regular Wireless Mesh Network (WMN) consists of mesh routers (MR)
that form a static wireless ad hoc network as an infrastructure and mesh clients (MC) that use that
infrastructure. As mesh networks are typically not stand alone networks, some of the mesh routers
function as gateways (GW) typically to the wired Internet. A subset of mesh routers function as
wireless access points (AP) where mobile mesh clients can connect to the network. The sets of
gateways and access points can overlap and they do not necessarily cover the entire set of mesh
routers.

![Wireless Mesh Networks](image)

Figure 1.1: Illustration of the Wireless Mesh Networks\cite{Askoxylakis et al., 2009}

The WMN, the cellular and WAP networks\footnote{By WAP networks, I mean wired networks where only the last links are wireless between the APs and the clients, and APs are connected to the infrastructure through wired links using e.g., regular Internet routers.} are mainly designed to serve as infrastructures. An AP of WAP networks provides high bandwidth, but small coverage. The equipments of WAP networks are cheap. In contrast to this, the cellular networks consist of expensive base stations which have low bandwidth, but large coverage. The main idea of WMNs is to combine the advantages of the cellular and WAP networks resulting in a network with similarly large coverage, but
1. INTRODUCTION

higher bandwidth than cellular networks have, but still consisting of cheap equipments such as in WAP networks. Note that in the case of WMN, the costs of the line installations can be saved, too, compared to WAP networks.

Due to the fact that the communications between the APs and MCs are wireless, MC can move during the data transmission. Meanwhile, they may run QoS aware applications. In order to be competitive with cellular networks, WMNs have to support user mobility and they have to fulfill QoS requirements, too.

The mesh routers are usually fixed nodes and have no power constraints, while, the computational performance of a usual mesh router can be described as mid-range devices. E.g., in my testbed I used mesh routers available in the market with a 32 bit ARM architecture based CPU on 175–800 MHz whose memory size varies between 16–128 Mb. The properties of mesh clients vary on a larger scale. A mesh client can be a sensor node, a smart phone, a laptop or even a desktop PC. They can be fixed or mobile, battery driven or powered.

The WMN concept can be applied to community based networks where the participants can share their resources through the WMN infrastructure. The concept can be utilized by the disaster recovery organization who can install mesh routers on their vehicle providing infrastructure to members during disaster recoveries. However, in this thesis, I concentrate on Wireless Mesh Networks, where the infrastructure is maintained by operators who provide broadband wireless access to the Internet for their customers based on contracts. The idea has gained increasing popularity (see e.g., Ozone’s mesh network in Paris (www.ozone.net) and the Cloud in London (www.thecloud.net)).

In such networks, a novel approach is that the mesh routers are operated by multiple operators, and they cooperate in the provision of networking services to the mesh clients. This cooperation can be based on business agreements (similarly to roaming agreements in the case of cellular networks). Customers may be associated with one or more operators by contractual means and have the ability to roam to the rest of the cooperating operators, if necessary.

The collaboration of multiple operators has many advantages. E.g., the installation cost can be reduced by using each other’s networking elements. Because the installation costs can be divided among the operators, the coverage can be increased to those places where the low number of potential users would not make it profitable for a single operator. Also, the spectrum can be utilized better, because the packet collision can be controlled easily within a single collaborating network in contrast to controlling the collision in independent overlapping networks.

The bandwidth capacity can be increased using multi-channel communication in WMN. Note that in single channel networks, the scalability is limited because the more radio range of wireless devices overlap, the higher the probability of packet collision is, which means that the throughput becomes less and less due to the higher packet loss rate. The multi-channel approach can increase the throughput of the wireless links, however, it decreases the connectivity of the routers. It is because those routers whose wireless interfaces are set to different channels are not able to communicate with each other even though they are within the radio range of each other. Using multiple wireless interfaces, MRs can set their interfaces such that they communicate with different neighbors using different channels. With this approach, the bandwidth can be increased while the connectivity of the network is still considerable.

I consider Wireless Mesh Networks, where the mesh clients connect to the access points directly (i.e., mesh clients are one hop away from the mesh network). In theory, mesh clients could provide data forwarding services to each other, and connect to the access points via multiple other mesh clients, but this would require special software on the mesh clients (essentially they would function as a router). Furthermore, this concept has special security requirements which are not considered in this thesis.

I refer to the above described Wireless Mesh Network, which is maintained by multiple operators, uses multiple interfaces with multiple channels, and supports user mobility and QoS, as Multi-WMN.
1. INTRODUCTION

1.2.2 Security issues

Attacker model

In order to be able efficiently protect the data forwarding mechanism in the mesh network, the potential attacker must be identified. Therefore, first, I classify the attackers and describe their objectives and their tools.

Classes of attackers  The attackers can be classified into the following three types:

- **Dishonest customers** are those mesh clients who have legitimate but restricted access to the network, and they want to gain illegal access.

- **Dishonest operators** are those who do not follow honestly the business agreements.

- **External attacker** has no internal access to the mesh network.

Objectives of attacks.  The main objectives of attacks can be the followings:

- **Unauthorized access to the services provided by the mesh network (e.g., Internet access):** External attackers may try to gain access to the mesh network without any subscription and dishonest customers may try to access services that are not included in their subscription. This type of attack makes the operators to forward messages that no one will pay for.

- **Unauthorized access to customer data and meta-data:** External attackers or dishonest operators may try to violate the confidentiality of the messages sent to or from the mesh clients or they try to violate the privacy of the mesh clients (e.g., customer’s location or service usage profile).

- **Denial-of-Service (DoS):** External adversaries may try to degrade the QoS level offered by the network or to completely disrupt the network.

- **Gaining advantage over competitors:** For dishonest operators, the primary reason to mount attacks on the system (especially on those parts that are operated by other operators) is to gain some advantage over their competitors. This is achieved either by destroying the reputation of a competitor, or by dishonestly increasing their own reputation.

Attack mechanisms.  All the above described objectives can be achieved by the combination of the following two sets of attack mechanisms:

- Attacks on wireless communications (including eavesdropping, jamming, replay, and injection of messages, and traffic analysis);

- Setting up fake mesh routers or compromising existing mesh routers (typically by physical tampering or logical break-in). The behavior of the fake or compromised mesh routers can be arbitrarily modified in order to help to achieve specific attack objectives.

Security requirements

Based on the adversary model described above, the following main security requirements can be identified for Multi-WMN in order to secure data forwarding mechanism:
1. INTRODUCTION

Authentication of mesh clients and access control. In order to prevent unauthorized access to services and unwanted traffic flow, mesh clients must be authenticated, and access control rules must be enforced in the system at the earliest point. It should take place at the access points. In the literature, there are many options which satisfies this specific requirement, however as I showed in the introduction of Wireless Mesh Networks, there are other requirements to satisfy in the Multi-WMNs, which may exclude some of those options. In brief, these requirements include the need to support end-user mobility and QoS-aware applications, and the need to work in a multi-operator environment.

QoS services may have requirements on the length of the interruptions in the communication that they can tolerate. When a mesh client moves from one access point to another, it has to re-authenticate itself as part of the handover process. Before a successful authentication process, the MC should not be allowed to access the network. Thus, the re-authentication delay must be minimized in order to ensure that the interruption caused by the handover remains tolerable for the applications. In addition, the multi-operator environment means that such handovers may occur between access points belonging to different administrative domains, and hence, the authentication and access control scheme must be able to handle this situation.

Protection of wireless communications. As I have already mentioned, wireless communication is vulnerable to eavesdropping, spoofing and replay attacks. In Wireless Mesh Networks, the wireless communication takes place not just between access points and mesh clients, but also among mesh routers. This extends the opportunities of the attackers. In order to minimize the effect of attacks the following considerations are suggested:

- Confidentiality and integrity of application data must be assured in an end-to-end manner in order to prevent unauthorized access to user data. This could be done between 1) the client application and the server, 2) mesh client and the gateway, or 3) access point and the gateway.

- In order to protect meta-data of the customers, not just the application data, but the end-to-end addressing must be hidden from external attackers, too. Link-by-link encryption of the header in case of presence of end-to-end encryption can help in this matter. It is favorable to maintain dummy traffic between idle links in order to prevent traffic analysis.

- In order to minimize the effect of fake, modified or replayed messages, they must be identified and dropped as early as possible. Therefore, the message integrity and authenticity must be verified in a link-by-link manner. This protection must be applied to control messages as well. Control messages can belong to e.g., neighbor discovery mechanism or routing protocol.

- As jamming cannot be prevented, the routing protocol must be robust against loss or deletion of some control messages.

Intrusion and misbehavior detection and recovery. In some situations, the proactive defense approach, such as the above described methods, are too complicated or impossible to implement at all. Such situation is the prevention of jamming or to prevent mesh routers to exhibit arbitrary Byzantine behavior. The latter attack can be performed by dishonest operators or by external attackers due to the lack of physical protection of mesh routers. Therefore, one must also consider the application of some reactive measures aiming at the detection and recovery from attacks based on intrusion and misbehavior.

As misbehavior can happen at any layer of the communication stack, misbehavior detection should be implemented in all layers. However, securing the routing protocol seems to be the most important requirement in this category, because interfering with the routing protocol may affect the entire network, whereas attacks at lower or upper layers seem to have a more limited effect. In particular, the effects of attacks at lower layers (e.g., on medium access control and channel assignment) are usually limited in space (i.e., localized), whereas the effects of attacks at upper
layers (e.g., at the transport or application layer) are limited to particular traffic flows in the network. Therefore, I will focus on securing the routing protocol.

Misbehaving routers have mainly three tools to perform attack on the routing protocol: 1) dropping data messages, 2) injecting data messages, or 3) lying about the metric information of their link or router capacity. There are other attacks too, such as modifying the metric information of other routers, but those attacks can be prevented by cryptographic means. In contrast to this, in case 2) and case 3), even if the message is fake or the metric information is invalid, the message itself can be authentic since the misbehaving router owns all the valid keys. However, misbehavior can be identified when data messages are forwarded.

Motivations for the three above described attacks are the followings: A router may drop data messages in order to gain advantage over competitors by dropping messages forwarded on behalf of other operators. A router may lie about its metric in order to redirect as much traffic as possible to itself. A router may inject fake data messages in order to degrade the QoS level.

1.2.3 Research objectives

So far, three groups of main issues have been addressed regarding secure data forwarding in Multi-WMN: 1) fast authentication of MCs and access control to network resources, 2) protection of wireless communication including secure routing, and 3) intrusion and misbehavior detection and recovery.

In this thesis, I address neither protection against jamming attacks, nor protection of wireless communications. Even the security in general of routing protocols is not considered. There are two issues addressed in this thesis regarding Multi-WMNs.

I concentrate on the authentication and access control mechanism. Recall that, the Multi-WMN is a QoS aware mesh network that is maintained by multiple operators, which cooperate in the provision of networking services to the mesh clients. In this context, the authentication delay has to be reduced, in order to support mobile users and seamless handover between the access points. Many proposed fast authentication schemes rely on trust models that are not appropriate in a multi-operator environment. In this thesis, my objective is to determine the requirements on authentication and access control methods in Multi-WMN and to propose one which satisfies all of them.

As I have already pointed out, the attacks against the routing protocols seem to be the most effective. Therefore, I address the problem of detecting misbehaving routers in Wireless Mesh Networks and avoiding them when selecting routes. The mesh routers can exhibit arbitrary Byzantine behavior by reprogramming their firmware. I assume that link-state routing is used, and misbehaving routers claim fake information about their link status or device properties. Note that misbehaving routers may hold valid keys, and the authenticity of their messages is assured, thus, the receiving routers may utilize this information. Current solutions suffer from high overload or they do not suit multi-channel communication environment. My main goal is to propose a misbehaving router detection mechanism which can identify those routers that send fake information about their link status and device properties. Furthermore, I want to avoid to overload the network, and I require to suit the multi-channel environment.

1.2.4 My achievements

In Chapter 4, I propose two authentication schemes: 1) a combination of HOKEY and IEEE 802.11r standards to suit multi-operator environment, and 2) a certificate-based authentication scheme that is investigated in details. I achieve a short authentication delay by requiring that the protocol is executed locally between the access point and the mesh client. I assume that the access point is always a constrained device, and I propose different mechanisms for mesh clients with different computational performance. For the challenging case when the mesh client has some constraints, I propose a novel mechanism where weak keys are used for digital signatures to decrease the latency of the authentication. The security of the weak keys is provided by short-term certificates issued by the owner of the key. I report on a prototype implementation of my proposed
1. INTRODUCTION

schemes and on the results of a detailed performance evaluation, where I compare my solution to
the currently available standard methods (e.g., EAP-TLS).

In Chapter 5, I essentially propose a reputation system, where trusted gateway nodes compute
Node Trust Values for the routers, which are fed back into the system and used in the route selection
procedure. The computation of the Node Trust Values is based on packet counters maintained
in association with each route and reported to the gateways by the routers in a regular manner.
The feedback mechanism is based on limited scope flooding. The received Node Trust Values
concerning a given router are aggregated, and the aggregate trust value of the router determines
the probability with which that router is kept in the topology graph used for route computation.
Hence, less trusted routers are excluded from the topology graph with higher probability, while the
route selection still runs on a weighted graph (where the weights are determined by the announced
link qualities), and it does not need to be changed. I evaluated the performance of my solution
by means of simulations. The results show that my proposed mechanism can detect misbehaving
routers reliably, and thanks to the feedback and the exclusion of the accused nodes from the route
selection, the number of packets dropped due to router misbehavior can be decreased considerably.
At the same time, my mechanism only slightly increases the average route length.
Stimulating cooperation in data dissemination using barter in Delay Tolerant Networks

2.1 Introduction

A potential problem in opportunistic and in delay-tolerant personal wireless networks is that the quality of the service provided by the system heavily depends on the users’ willingness to cooperate. In particular, the users may act selfishly meaning that they download messages from other users that are interesting for them, but they deny storing and distributing messages for the benefit of other users. As shown in [Panagakis et al., 2007], if the majority of the users behave selfishly, then the message delivery rate decreases considerably and the quality of service provided by the network decreases accordingly.

The problems identified in [Panagakis et al., 2007] are the motivation for proposing a mechanism that encourages the users to carry other users’ messages even if they are not directly interested in those messages. My proposed mechanism is based on the principles of barter: the users trade in messages and a user can download a message from another user if he/she can give a message in return. I expect that it is worth for the users collecting messages even if they are not interested in them to exchange them later for messages that they are interested in. Thus, the messages are expected to disseminate faster in the network.

I analyze my proposed solution using game-theoretic techniques. I show that it is worth for the users collecting and disseminating messages even if they are not interested in them, which means that my approach indeed discourages selfishness. In addition, the results show that, in practical scenarios, the message delivery rate considerably increases, if the mobile nodes follow the Nash Equilibrium strategy in the proposed mechanism compared to the data dissemination protocol when no encouraging mechanism is present.

The idea of using barter mechanism in order to motivate selfish nodes to disseminate messages has been published in [Buttyán et al., 2007a] with preliminary investigation, and in [Buttyán et al., 2010a] with extended analysis.

This chapter is organized as follows. I summarize the related work in Section 2.2. In Section 2.3, I analyze the system without any incentives and determine the scenarios where stimulating mechanism should be introduced. In the same section, I introduce the system model that is used to analyze the system with and without incentives. I describe my barter based approach, and I also extend the system model with the barter mechanism in Section 2.4. For the analysis of the effects of selfish behavior on the system augmented with the barter mechanism, I introduce a game-theoretic model in Section 2.5. In Section 2.6, I show and interpret the results of the barter game. I describe
2. STIMULATING COOPERATION USING BARTER IN DELAY TOLERANT NETWORKS

in Section 2.7 how my proposal or model can be improved in the future. Finally, I give a summary of this chapter in Section 2.8.

2.2 State-of-the-art

So far, the problem of selfish nodes has been addressed mainly in the context of mobile ad-hoc networks and peer-to-peer file-sharing. Regarding the mobile ad-hoc networks, the proposed solutions to stimulate cooperation can be broadly classified into two categories: reputation systems and virtual payment based methods. Several researchers proposed reputation systems for ad hoc networks [Michiardi and Molva, 2002; Buchegger and Boudec, 2002], and in [Voss et al., 2005], an opportunistic solution is presented. For the virtual payment based methods, some proposed solutions can be found in [Buttyán and Hubaux, 2003; Zhong et al., 2003] in traditional ad-hoc networks and there are opportunistic network specific solutions in [Önen et al., 2007]. Usually, these solutions require authentication (and related key management), and/or the presence of a trusted third party. In addition, the payment based solution also raises the problem of determining the price of different actions (see e.g., [Crowcroft et al., 2004]).

Researchers have also studied under what conditions cooperation can emerge spontaneously among the nodes in ad-hoc networks (see e.g., [Srinivasan et al., 2003; Félegyházi et al., 2006]).

In peer-to-peer file-sharing systems, the researchers faced to the problem of freeriding. Freeriders are the users who try to download files from the others, but they do not share or upload anything. There are many solutions arisen in the last years (e.g., Kazaa, eMule, Gnutella [Adar and Huberman, 2000; Porter and Shoham, 2004]), but the most efficient one is the BitTorrent [Cohen, 2003; Neglia et al., 2007]. BitTorrent uses tit-for-tat to motivate the users to share and upload files. The solution is so successful that in some cases the users download files on behalf of other users. In the current work, I adapt the tit-for-tat in order to motivate the users in DTN to carry messages even if they are not interested in.

The application of delay-tolerant networks for personal wireless communications is considered in [Karlsson et al., 2006]. In particular, the authors show, by analytical tools and by means of simulations, that delay-tolerant networks can achieve a reasonably high throughput such that they can support various personal communication services.

In [Panagakis et al., 2007], the authors raise the problem of selfishness in delay tolerant networks. The authors study the performance of three representative routing algorithms in the presence of some selfish nodes. They show that when the nodes behave selfishly, the performance decreases, in the sense that messages are delivered with a longer delay if they are delivered at all. However, the authors do not propose any mechanism to stimulate cooperation. The results presented in [Panagakis et al., 2007], can be viewed as a motivation for my work.

In [Koukoutsidis et al., 2008], the authors considered the same subject. They have proven by analytical tools that the most beneficial behavior is to follow a forwarding strategy that the mobile nodes agreed on before. The forwarding strategy is described by the probability of forwarding uninterested messages. In contrast to the above mentioned analytical model, I investigate a more complex model.

The barter mechanism was introduced first and analyzed by a preliminary model in [Buttyán et al., 2007a]. The most important new contributions are that I pin-pointed the scenarios where encouraging mechanism is required and I showed that the barter based mechanism increases data delivery in a more general and realistic model. Furthermore, I presented a more exhaustive analysis of the results.

2.3 System analysis

In this section, I introduce my system model, which is general enough to represent different applications, and it is particularly well adapted for the applications described in Section 1.1. Because of the complexity of the model, I use simulations instead of analytical tools. I show that there are
scenarios where the message delivery has large latency because the mobile nodes are selfish in a sense that they only store and forward messages that they are directly interested in. The aim of the analysis is twofold: 1) to see whether an incentive is required in the network to increase the message delivery rate and decrease the message delivery latency, and 2) to give a reference with which I can compare my subsequent solution.

2.3.1 System model

In my model, the users are placed in an arbitrary field. They own devices that have capabilities to communicate with other devices within their radio range. I consider the case when the devices communicate via wireless links, however, or analysis can be extended to wired communication too. The used wireless technology can be Bluetooth, Wi-Fi or any suitable wireless techniques. The messages are generated and disseminated among the devices/users in the considered system, but each user is interested only in a small subset of the messages. The dissemination process is based on the store-carry-and-forward principle. A user and her device together is the mobile node, and I assume that the message destination has no impact on the user’s movement.

Each message has a type for each mobile node. For simplicity, I distinguish only two types: primary messages and secondary messages. A message is a primary message for a given mobile node, if the mobile node is interested in the content of the message and secondary if the mobile node is not. Note that a message may have different types for different mobile nodes, as different mobile nodes are interested in different contents.

The content of the messages can be anything: sensor data (e.g., temperature, humidity), piece of text (e.g., advertisement, local information). These messages are generated by special nodes which I call message nodes. In my system model the time is slotted, and the message nodes generate new messages with a fixed average rate: \( \varrho \) messages per time step. In the considered scenarios, the message nodes are typically constraint devices (e.g., sensors) which do not communicate with each other but with the nearby mobile nodes because they have not enough memory to maintain routing tables. The message nodes are static and due to the memory constraints, each one stores only the most recently generated message, which can be downloaded at the cost of communication by any mobile node that passes by the message node.

A message has two main properties: the first one is the *popularity* attribute and the second one is the *discounting characteristic*. The popularity attribute \( 0 < \zeta \leq 1 \) describes the probability that a randomly taken mobile node is interested in the message. I assume that message nodes do not generate irrelevant messages, hence I consider \( \zeta > 0 \).

Each message has some value for each mobile node. The value of a message is determined by its age. For simplicity, I assume that primary messages of the same age have the same value for the mobile nodes. Without loss of generality, I assume that the value of a primary message at the time of its generation is one unit, and this is discounted in time, because messages lose their value over time. This is usually the case in the applications that opportunistic networks are envisioned for. The discounting characteristic is described with a function: \( \delta(t) \). The discounting function determines the value of the messages over time. Obviously, it is difficult or impossible to find a discounting function which suits each application. Therefore, I defined three different monotonely decreasing discounting functions. I express these function in Eqs. (2.1)–(2.3) and I plot them in Figure 2.1. In the first case, the message value decreases linearly, in the second case, the messages devaluate exponentially, and in the last case, the messages lose their value suddenly, similarly to a step function.

\[
\delta_0(t) = \begin{cases} 
1 - \frac{t}{500} & \text{if } t < 500 \\
0 & \text{else} 
\end{cases} 
\]  
(2.1)

\[
\delta_1(t) = 0.995^t 
\]  
(2.2)

\[
\delta_2(t) = 1 - \frac{1}{1 + 1000 \cdot (1 - \frac{1}{20})^t} 
\]  
(2.3)
When two mobile nodes get in the vicinity of each other, they interact in the following way:

1. The mobile nodes exchange the list of the messages that they carry. The exchanged lists contain only the short descriptions of the messages (including their time of generation) rather than the messages themselves.

2. Each mobile node $u$ removes from the list $L_v^{(0)}$ received from $v$ the messages that are not primary for node $u$, and the ones that $u$ already stores in memory getting the list $L_v^{(1)}$.

3. Each mobile node $u$ determines the value of the messages listed in $L_v^{(1)}$ based on their ages. Then, each mobile node orders the messages contained in $L_v^{(1)}$ by their value in descending order. The resulting ordered list $L_v^{(2)}$ is the list of messages that $u$ wishes to download from $v$.

4. Mobile nodes $u$ and $v$ download messages from each other following the lists $L_v^{(2)}$ and $L_u^{(2)}$, respectively, until they move out from each other’s radio range.

Connections can be interrupted because the nodes are mobile and they leave the radio range of the other party. Therefore, in my model, the mobile nodes are not able to exchange as many messages as they want but at maximum one message per time step. Hereby, I assume that a message exchange is either completed in the time step or not started at all. This limited exchange capability is called the *implicit cost* of the exchange, because there is no guarantee that the nodes can download all the messages that they want from the other party.

In my system model, there is no other cost. In a scenario that I imagine the communication cost is negligible as the battery of the personal devices can be recharged easily day by day. The storage cost has two aspects: 1) The messages need storage space and storage constraint may limit the number of stored messages. This limitation is not significant as the storage space required for storing the data downloaded by using wireless technology is less than the memories offer, nowadays. 2) The time needed to determine which messages and in what order the nodes want to download increases polynomially with the number of message stored by the other party. To control this, the mobile nodes delete the valueless messages, thus, they delete the messages from the memory whose value goes below a certain threshold $D$, $0 < D < 1$.

To measure the message delivery rate and delivery latency, I define a formula for the goodput (see Eqs. (2.4) and (2.5)). The notation is the following considering mobile node $i$:

- $m_i^t$ is the message that mobile node $i$ downloaded in time step $t$
- $T_m$ is the time step when message $m$ was generated
2. STIMULATING COOPERATION USING BARTER IN DELAY TOLERANT NETWORKS

Figure 2.2: The convergence of the goodput of some sample nodes

- $\delta$ is the discounting function described above
- $v_i(t)$ is the gain that mobile node $i$ gets in time step $t$, and it is defined as follows:

$$v_i(t) = \delta(t - T_{m_i})$$  \hspace{1cm} (2.4)

Let $M_P^i(t)$ denote set of messages that were generated until time $t$ and are primary for node $i$. The cardinality of $M_P^i(t)$ describes the maximum value that node $i$ can obtain until time $t$ as the value of each message is 1 at the moment of the generation. As shown in Eq. (2.5), the goodput ($0 \leq G_i(t) \leq 1$) for mobile node $i$ at time $t$ is the sum of the value of each primary message at the time of obtaining normalized with the value that node $i$ could obtain in an ideal case.

$$G_i(t) = \frac{\sum_{\tau=0}^{t} v_i(\tau)}{|M_P^i(t)|}$$  \hspace{1cm} (2.5)

Note that the goodput is time and mobile node specific. However, the distribution of $G_i$ is the same for each mobile node $i$ if all the mobile nodes behave in the same way. The goodput may vary over time, however as I show in the next subsection, the value of the goodput converges to a steady-state value. Therefore, I will consider the goodput, denoted by $G_i$, of each mobile node $i$ in the steady state.

$$G_i = \lim_{t \to \infty} G_i(t)$$  \hspace{1cm} (2.6)

2.3.2 Convergence of the goodput

I built a simulation environment where I investigated the selfishness of the mobile nodes. This simulation environment is described in the following sections. However, I present here two figures showing that the goodput of the nodes converges to a limiting value. This can be seen in Figure 2.2 where the goodput of some randomly chosen mobile nodes is plotted against the time. In Figure 2.3, the average goodput and its deviation of all mobile nodes is plotted against the time. In this section, I prove by analytical tools that the goodput obtained after a fixed number of time steps in simulation close to the steady-state goodput.

The considered system model can be represented as a finite state Markovian model. A state of the Markovian model can be described at time $t$ as Eq. (2.7) shows.
The convergence of the average goodput and its empirical deviation

\[ s(t) = \{B_1(t), B_2(t), \ldots, B_N(t), \]
\[ Z_1(t), Z_2(t), \ldots, Z_N(t), \]
\[ H_1(t), H_2(t), \ldots, H_N(t)\} \]

where

- \(N\) is the number of nodes
- \(B_i(t) = [m_{i1}, m_{i2}, \ldots]\) is the buffer of node \(i\), where the messages are stored.
- \(Z_i(t) \in \{*, m\}\) is message stored in the memory of the message node \(i\), where * denotes the case when no message is stored at time \(t\), otherwise \(m\) stands for the generated message, which arrives from the — in principle — infinite space of messages.
- \(H_i(t)\) is the position of node \(i\) on field \(F\) where mobile nodes move.

As one can see the state space of the Markovian model described above is infinite, however with some feasible assumptions the model can be converted to a finite state model.

- Note that the memory of mobile nodes was assumed to be unlimited, however an upper bound can be defined. Recall that the mobile nodes delete the messages if the message is older than \(T\) time step. Let the number of message nodes be \(n\). The greatest number of messages is generated if all the message nodes generate a new message in each time step. A message disappear from the system after \(T\) time steps. Therefore, the greatest number of messages that a node may store is \(L = n \cdot T\). Hereby, \(B_i(t) = [m_{i1}, m_{i2}, \ldots, m_{iL}]\).

- In the Markovian model described above, each message \(m\) arrives from infinite space as there was no restriction for it. However, it is feasible to assume that the length of the digital contents that the nodes exchange is limited, let us assume to be \(l\) bits. In that case, the size of the message space is \(2^l\).

Note, that the state space can be described by a deterministic mapping as Eq. (2.8) shows.

\[ s(t + 1) = f[s(t), \]
\[ r_1(t + 1), r_2(t + 1), \ldots, r_N(t + 1), \]
\[ r'_1(t + 1), r'_2(t + 1), \ldots, r'_M(t + 1), \]
\[ r''_1(t + 1), r''_2(t + 1), \ldots, r''_M(t + 1)] \]
where

- $r_i(t+1)$ is a random element used as an input by the algorithm to calculate the next step of node $i$ ($1 \leq i \leq N$) on the field $F$ at time $t + 1$.
- $r_j^*(t+1)$ is a random element used as an input of message generation of message node $j$ ($1 \leq j \leq n$) at time $t + 1$.
- $r_k^*(t+1)$ is a random element used as an input of node pairing in meeting point $i$ ($1 \leq k \leq M$).

The random numbers are generated independently of the time and of each other.

Note, that the state transition mapping is time independent. The sequence of state random variables $S(0), S(1), \ldots, S(t), \ldots$ constitutes a discrete time homogenous Markovian chain. The transition matrix of the Markovian process can be derived from Eqs. (2.7) and (2.8). An element $P_{ij}$ of the transition matrix describes the probability that the system from state $i$ change to state $j$.

A Markovian chain is *ergodic*, if following limiting values exist:

$$\lim_{n \rightarrow \infty} P_{ik}^{(n)} = P_k$$

the limiting values are independent of $i$ and

$$\sum_{k=1}^{\infty} P_k = 1$$

As the classic theorem of Markovian chains claims, a finite state homogenous Markovian chain is ergodic, if it is irreducible and aperiodic. Particularly, there is a time step $t$ and a state $j$, such that state $j$ can be reached from arbitrary initial state $i$ with positive probability with time step $t$. The convergence to limiting distribution $P_j$ is exponential, which means the following: let $P_{ij}^{(t)}$ denote the probability, that the Markovian chain starting from state $i$ arrives at state $j$ with $t$ steps, furthermore, let denote the stationary probability of state $j$ by $P_j$, the difference $\vert P_{ij}^{(t)} - P_j \vert$ decreases exponentially when $t$ tends to infinity (Theorem of Markov). In this case, uniform exponential bound exists for difference $\vert P_{ij}^{(t)} - P_j \vert$ independently of $j$.

In my model, the proof of the condition for ergodicity is the following: Assume the system is in an arbitrary state. I select a state $k$, let this state be the one where the buffer of the first node contains a single fresh message, while all other buffers are empty. Such a state can be reached from any other states in the following way: First, the buffers of the nodes become empty such that the users move or stagnate at a fixed position without meeting any message generator nodes. As the time passes the aging messages are dropped out from the buffer. Then the first node approaches a message node where it generates a fresh message and the node receives that message.

As it is shown above, my system is ergodic. Thus, the distribution of the stationary state is approached at exponential rate.

Considering Eq. (2.5), the goodput is affected by the transient state of the system also, not just on the stationary state. However, from the ergodicity of the Markovian chain, it follows that the effect of the transient state becomes negligible and fades away at an exponential rate if the time goes to infinity. By empirical observation, it is appropriate to consider the goodput after time step 3000 as the goodput will not change in the future considerably.

**2.3.3 Simulations**

In order to investigate the selfishness of the mobile nodes, I built a simulation environment in C++ where the fixed-number of mobile nodes move in discrete time steps according to one of the two mobility models: the Restricted Random Waypoint (RRW, [Blažević et al., 2002]) and Simulation of Urban MOibility (SUMO, [SUMO, 2010]) model.
In the restricted random waypoint model, 300 mobile nodes move on a field of size $20 \times 20$ unit initially placed uniformly at random. The random waypoint model is characterized as restricted, because the mobile nodes are not allowed to choose any point as a destination, but some appointed places on the field chosen at random, called meeting points. Each mobile node selects a meeting point randomly, and moves towards this meeting point along a straight line with a fixed speed. When the meeting point is reached, the mobile node stops and stays for randomly chosen time (10 time steps on average). Then, it chooses another meeting point and begins to move again. The nodes that happen to be at the same meeting point in the same time step are paired randomly and these pairs are able to download one message from each other in the above described way.

Figure 2.4: Simplified map of Budapest used in SUMO mobility model [Buttyán et al., 2007b]

SUMO is an open source, realistic road traffic simulator. 300 vehicles (mobile nodes) start their movement from a randomly chosen place at a randomly chosen time and they follow the traffic rules moving towards their destination also chosen at random in a predefined map. I implemented a simplified map of Budapest, Hungary with 60 intersections (including the dead ends) in SUMO as shown in Figure 2.4 and the vehicles move on this map. In each edge, there is a speed limit specified calculated automatically by the SUMO (in most cases 35 m/time step and sometimes 40 m/time step). The vehicles accelerate, move constantly at the highest speed, slow down and stop depending on the traffic. The nodes can communicate with each other when they stop in the intersections similarly to the meeting points in the restricted random waypoint model. The vehicles leave the meeting point as soon as the traffic admits.

In Figure 2.5, I compare the two considered mobility model with respect to the duration of getting from a meeting point to another neighboring one. Two meeting points ($A$ and $B$) are neighbors if a mobile node can go from $A$ to $B$ and back without stopping at any other intermediate meeting points. Note that in the restricted random waypoint model, any two meeting points are neighbors, but in the SUMO, only those meeting points that are linked in Figure 2.4. In Figure 2.5(a) and 2.5(b), the histogram of the time steps needed to reach a meeting point is shown. In the case of the SUMO, the duration was specified considering the maximum speed of the cars. As a comparison I can state that in the restricted random waypoint mobility model the mobile nodes communicate with each other more frequently than in the case of SUMO.

Recall that in my system model, the messages are injected into the network by message nodes
that are static. In the restricted random waypoint model, the message nodes reside in the meeting points, whereas in SUMO the message nodes are placed in each intersection.

As I have already described each message has a popularity value $\zeta$. In the simulation, when a message node generates a new message, it determines which mobile node is interested in it according to the popularity value. Thus, the message node sets the message to primary with probability $\zeta$ for each mobile node. With this, I can prevent in the simulations that sometimes a node indicates a message as primary and sometimes as secondary. All the message nodes together generate one new message per time step on average both in case of SUMO and restricted random waypoint model.

In each simulations, all the messages have the same discounting characteristic, one of the function described in Section 2.3.1 (see Eqs. (2.1)–(2.3)).

I determine the length (number of time steps) of the simulation in an empirical way by taking into account that the goodput have to reach the steady-state goodput. When I run the simulations for 3000 time steps, the average goodput have not changed considerably for 1000 time steps in the analyzed simulations. Therefore I run all simulations for 3000 time steps.

I summarize the simulation parameters in Table 2.1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>RRW</th>
<th>SUMO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation length (time steps)</td>
<td>3000</td>
<td></td>
</tr>
<tr>
<td>Number of mobile nodes</td>
<td></td>
<td>300</td>
</tr>
<tr>
<td>Number of meeting/cross points</td>
<td>100</td>
<td>60</td>
</tr>
<tr>
<td>Number of message nodes</td>
<td>100</td>
<td>60</td>
</tr>
<tr>
<td>Message generation rate $\varrho$</td>
<td>0.01</td>
<td>0.0166</td>
</tr>
<tr>
<td>Simulation area</td>
<td>$20 \times 20$ unit</td>
<td>see Fig. 2.4</td>
</tr>
<tr>
<td>Velocity (unit/time step)</td>
<td>1</td>
<td>induced by</td>
</tr>
<tr>
<td>Probability of leaving a meeting point</td>
<td>0.1</td>
<td>SUMO</td>
</tr>
<tr>
<td>Threshold for message erase $D$</td>
<td>0.05</td>
<td></td>
</tr>
</tbody>
</table>

I varied some of the parameters to study their effect on the results. As described above during simulation runs, I used different functions for message devaluation. Besides this, for the sake of simplicity, I assumed that during a simulation the messages are generated with one pre-defined popularity attribute $\zeta$, but I executed more simulations with different $\zeta$ values. Recall
that $0 < \zeta \leq 1$. To reduce the complexity of my simulations, I use the following values of $\zeta$: $\zeta = 0.05, 0.2, 0.4, 0.6, 0.8, 1$.

The main objective of these initial simulations is to pin-point the circumstances when an incentive mechanism is required to increase the message delivery rate and to decrease the message delivery latency. Therefore, I run two kinds of simulations for every scenario: 1) one to get the goodput when the nodes behave selfishly and 2) another one to get an upper bound for the goodput. In the former case, the mobile nodes strictly follow the protocol introduced in Section 2.3.1. This protocol corresponds to selfish behavior, because the mobile nodes download only those messages in which they are interested. To get an upper bound for goodput, the mobile nodes download all the new messages that they find in the memory of the connected node in one time step, both the primary and secondary ones. Clearly, this upper bound is different from the theoretical maximum of 1, because the value of a message decreases before reaching an interested mobile node, if reaches it at all.

As I have already stated, the distribution of the goodput achieved by the mobile nodes is the same. Therefore, I determine the goodput of the network by getting the average goodput of all the nodes.

2.3.4 Conclusion

Results in the case of the restricted random waypoint model and SUMO can be seen in the Figures 2.6(a) and 2.6(b), respectively. In these figures, I show simulations where the discounting function is linear ($\delta_0$), because the results show minor changes with other message devaluations.

In these figures, the goodput of the network is plotted against the popularity attribute value of the messages. To remind the reader, in the simulations in each parameter set, the messages have the same popularity value. The solid line shows an upper bound for the goodput and the line with dashes and dots shows the goodput of the network in the selfish case, when the mobile nodes do not download secondary messages. I present the 95% confidence intervals at each simulation points.

There are significant differences between the two mobility models. In the case of the restricted random waypoint model (shown in Figure 2.6(a)) the goodput is much higher than the one in the SUMO mobility model (shown in Figure 2.6(b)). This difference has two reasons:

- In SUMO mobility model, the traffic is higher at the central meeting points than in the
suburb as it is the case in all the cities. Where the traffic is low, the mobile nodes can quickly bypass the message nodes. For this reason, the messages generated there may be deleted before passing to any mobile node. Recall that a message node can store only one message. Therefore, the message node overwrites a message if a new one is generated.

- As Figure 2.5 shows, in the case of SUMO mobility model, the distances between the meeting points are longer than in the case of restricted random waypoint model. Recall that mobile nodes are able to exchange messages only while they do not move. Furthermore, in the SUMO, the mobile nodes can bypass quicker the meeting points than in the restricted random waypoint model. All in all, the mobile nodes have less opportunity to exchange messages in the case of SUMO.

When the mobile nodes behave selfishly the popularity value has a large impact on the goodput. The more mobile nodes are interested in a message, the more nodes download the message even if all the mobile nodes are selfish. The more mobile nodes download a message, the higher is the probability that a mobile node will meet one who has already downloaded the message. I call this the selfish carrier effect and it can be seen in the Figure 2.6(b), but not clearly in the Figure 2.6(a). There, the goodput increases with the increasing popularity until a specific value, but then the goodput decreases.

The reason for the decrease of the goodput while the popularity increases is the following: The goodput is a ratio as Eq. (2.5) shows. As one can see, the denominator (maximum value) can increase to infinity. While the numerator (obtained value) has an upper bound (even if it is difficult to determine in a concrete parameter set), because the nodes are able to exchange only one message in each time step. Thus, if the number of the interested messages increases, but the obtained value reached its upper limit, then the goodput decreases.

As a conclusion, I can state that the goodput is affected by two mainly independent, but opposite effects: the selfish carrier effect and the implicit cost. When the value of the popularity attribute is 1 the goodput is affected mainly by the implicit cost, whereas when the popularity value is near to 0 it is affected clearly by the selfish carriers. The implicit cost comes from a property of the system model, while the selfish carrier effect comes from the selfishness of the mobile nodes. Therefore, I can state that an incentive is required to compensate the selfish carrier effect which mainly affects the goodput of the network when the popularity value of the generated messages is low.

### 2.4 My approach

My approach to stimulate the cooperation of mobile nodes is based on the principles of barter. More specifically, as mentioned above when two nearby mobile nodes establish a connection, they first send the description of the messages that they currently store to each other, and then they agree on which subset of the messages they want to download from each other. In order to ensure fairness, the selected subsets must have the same size, and the messages are exchanged in a message-by-message manner, in preference order. If any party cheats, the exchange can be disrupted, and the honest party does not suffer any major disadvantage (i.e., the number of messages downloaded by the honest party is at most one less than the number of messages downloaded by the misbehaving party).

Note that it is entirely up to the mobile nodes to decide which messages they want to download from each other. They may behave selfishly by downloading only those messages that are of primary interest for them. However, selfish behavior may not be beneficial in the long run. In particular, the idea is that a message that is not interesting for a mobile node A may be interesting for another mobile node B, and A may use it to obtain a message from B that is indeed interesting for A. In other words, the messages that are secondary for a mobile node still represent a barter value for the mobile node, and hence, it may be worth downloading and carrying them. Thus, the messages can be viewed as an investment to get new primary messages later.
Recall that the selfish mobile nodes ignore the secondary messages when they selected the messages to download in the message exchange protocol introduced in Section 2.3.1. However, when the messages are exchanged according to the principles of barter, as it is mentioned above, it is worth downloading and carrying secondary messages too, even if the mobile nodes are selfish (I will show that this statement holds). But, the mobile nodes have to compare the value of primary to the value of the secondary messages when they select which messages and in what order they want to download from the connected party.

Recall that there is no direct benefit of downloading a secondary message. It is worth to download to exchange later for primary ones. According to this, the value of the secondary messages is considered only when a node sorts the messages for downloading from another node. The value of a secondary message at the time of its generation depends on how the mobile node values secondary messages with respect to primary messages. The secondary value is discounted in the same way as primary messages. In other words, if for a mobile node, secondary messages are worth \( SP \) units for some \( 0 \leq SP \leq 1 \) at the time of their generation, then the value of a secondary message after \( t \) time units is \( SP \cdot \delta(t) \). \( SP \) is called secondary/primary ratio. I have to emphasize that if \( SP_u = 0 \) then the mobile node \( u \) does not download any secondary messages. I have chosen ratio \( SP \) as the strategy of the players. This ratio intuitively represents the level of cooperativeness of the node: the higher this ratio is, the more the node values secondary messages, hence, the higher the number of the secondary messages (not directly interesting for the node) that are downloaded and carried by the node. An advantage of this modeling choice is that representing the strategy with a single real valued parameter was easy to handle in the simulation.

Note that in general, the value of a secondary message cannot be larger than the value of a primary message of the same age (i.e., \( SP \leq 1 \)), because the primary message has the same barter value as the secondary message, and in addition, the mobile node is interested in its content. However a specific secondary message which is more fresh than a specific primary message may have higher value and it can be exchanged for primary messages later, which will have higher gain all together.

I adapt the message exchange protocol according to the barter-based approach in the following way:

1. The mobile nodes exchange the list of the messages that they carry.
2. Each mobile node \( u \) removes from the list \( L_u^{(0)} \) received from \( v \) the messages that \( u \) already stores in memory, and thereby obtains the list \( L_u^{(1)} \).
3. Each mobile node \( u \) determines the value of the messages listed in \( L_u^{(1)} \). The value is \( \delta(t) \) if the message is primary and \( SP_u \cdot \delta(t) \) if the message is secondary (\( t \) denotes the age of the message). The list obtained in this way is denoted by \( L_u^{(2)} \).
4. Each mobile node \( u \) orders the messages contained in \( L_u^{(2)} \) by their value in descending order. The resulting ordered list \( L_u^{(3)} \) is the list of messages that \( u \) wishes to download from \( v \).
5. The nodes exchange at most \( \ell = \min(|L_u^{(2)}|, |L_v^{(2)}|) \) messages from the beginning of their lists on a message-by-message manner, where \( |L| \) denotes the length of the list \( L \). Thus, the number of exchanged messages is determined by the length of the shorter list or the duration of the connection.

I assume that the mobile nodes offer all their valid and only valid messages to download. It is not worth for any nodes to hide messages from other mobile nodes, because it may decrease the number of messages that the mobile node is allowed to download from other mobile nodes. In addition, I assume that a mechanism is present in the system that prevents injecting fake messages. This is important, because greedy nodes can increase the number of messages that they can offer by injecting fake messages.

In order to show that the latter assumption is feasible, I sketch the operation of two mechanisms that would prevent the injection of fake messages:
• One prevention mechanism can be based on digital signatures. The mobile nodes are allowed to exchange only those messages that have a valid digital signature. The digital signatures are added to the messages by an authority. This authority can be represented by the message nodes and in that case, the message nodes are responsible not just for generating the messages but certifying them, too. Although, this solution filters out the fake messages it may not be applicable in some application.

• Another mechanism for preventing injections of fake messages can be based on a reputation mechanism. The users can define a threshold and they download only messages whose reputation value is higher than the threshold and the users can evaluate the messages or the services which generates the messages themselves. The evaluation messages may be distributed among the mobile nodes. Note that this reputation mechanism is not related to the mobile nodes’ willingness of the message distribution, but it refers to the quality of the message contents. This kind of reputation mechanism can complement my barter mechanism.

The purpose of my analysis later in this chapter is to verify whether the barter based approach indeed increases the goodput.

2.5 Game model

I model my proposed mechanism as a game to analyze the behavior of the mobile nodes using game-theory [Félegyházi and Hubaux, 2006; Fudenberg and Tirole, 1991; Gibbons, 1992; Osborne and Rubinstein, 1994]. My objective is to investigate whether the network can reach high goodput using barter mechanism even if selfish mobile nodes are present.

I define a non-cooperative game \( G = [P, \{S_i\}, \{\pi_i\}] \), called barter game. \( P \) is the set of the players, \( S_i \) denotes the strategy space of player \( i \in P \), and \( \pi_i \) represents the payoff function of each player \( i \). To be more precise, \( \pi_i \) is the simplified notation of \( \pi_i(s_0, s_1, \ldots, s_{|P|-1}) \), because the payoff of each player depends on the strategy played by the other players. This can also be denoted by \( \pi_i(s_i, s_{-i}) \) emphasizing the strategy of player \( i \), where \( s_{-i} \) is the strategy profile of all the players except for player \( i \).

In the barter game, the players \( P \) are the mobile nodes, and hence in the rest of this chapter, I will use the same notation for players as for mobile nodes. The strategy of each player is its secondary/primary ratio \( (SP_i \in S_i = [0,1]) \). The players do not change their strategies during the game. The players choose their strategies in a way to maximize their goodput. Hence, the steady-state goodput is the payoff of the barter game for player \( i \).

\[
\pi_i = G_i
\]  

(2.9)

In order to model the behavior of the selfish mobile nodes, I introduce the concept of best response and Nash Equilibrium [Fudenberg and Tirole, 1991; Gibbons, 1992; Osborne and Rubinstein, 1994].

The best response of player \( i \) to the profile \( s_{-i} \) is a strategy such that:

\[
B_i(s_{-i}) = \arg \max_{s_i \in S_i} \pi_i(s_i, s_{-i})
\]  

(2.10)

If player \( i \) plays strategy \( B_i(s_{-i}) \), it reaches the maximum from the obtainable payoffs given that the other players play \( s_{-i} \).

The pure-strategy profile \( s^* \) is a Nash Equilibrium if the following equation holds for \( s^* \):

\[
s^*_i = B_i(s^*_{-i}), \forall i \in P
\]  

(2.11)

Namely, in Nash Equilibria none of the players can increase their payoff by changing their strategy unilaterally.

A game \( G = [P, \{S_i\}, \{\pi_i\}] \) is symmetric if each player has the same strategy space \( (S_0 = S_1 = \ldots = S) \) and their payoff functions are equal \( (\pi_i(s_i, s_{-i}) = \pi_j(s_j, s_{-j})) \) for \( s_i = s_j \) and \( s_{-i} = s_{-j} \), where \( i, j \in P \). A symmetric game \( G \) can be denoted by \( [P, S, \pi()] \).
As one can see, the barter game is a symmetric game, because the strategy space defined in the game is identical for all players. In my system model, the nodes are not distinguished. Thus, they can maximize their payoff in the same way and they get the same payoff in the same strategy profile.

In the analysis of the barter mechanism, I am looking for the Nash Equilibria. I limited ourselves to find only pure strategy, symmetric Nash Equilibria. This is because, I assumed that each mobile node is a player, which leads to the analysis of a game with a \(|P|\)-dimensional strategy space. The exhaustive analysis of the entire strategy space is thus infeasible by means of simulations.

A symmetric game has symmetric pure-strategy equilibria [Cheng et al., 2004], if the strategy space is a nonempty, convex and compact subset of some Euclidean space while the function of payoff is continuous in the strategy and quasiconcave. In my case, the strategy space is the interval \([0,1]\), which corresponds to the conditions of existing symmetric pure-strategy equilibrium. Whereas, the properties of the payoff function are not verifiable, the results of the simulations will show that the conditions hold.

If I expand (2.10) and (2.11) according to the symmetric game and equilibrium, \(\{s^*\}\) is Nash Equilibrium if the following equation holds for any player \(i \in P\):

\[
s_i^* = \arg \max_{s_i \in S} \pi(s_0^*, s_1^*, \ldots, s_i, \ldots), \text{where } s_u^* = s_v^* \forall u, v \in P/\{i\} \quad (2.12)
\]

As one can see, it is easy to verify that a specific strategy profile \(\{s'\}\) is a Nash Equilibrium or not. Considering any player \(i \in P\) — without loss of generality \(i = 0\), called player NULL — if it is worth for player \(i\) to deviate, \(\{s'\}\) is not a Nash Equilibrium, whereas if \(s'\) is the best response to player \(i\) then \(s'\) will be the best response strategy for all the other players too, as the players have equal payoff functions.

Therefore, to find the symmetric pure-strategy Nash Equilibria, it is not necessary to examine the whole \(|P|\)-dimensional strategy space, but investigation of a 2-dimensional space is enough. In order to find all the symmetric pure-strategy Nash Equilibria, I consider all the symmetric pure-strategies \(\{s'\}\) as Nash Equilibria candidates. Then, I consider the whole strategy space that player NULL can play to check if a Nash Equilibrium candidate is indeed a Nash Equilibrium or not. The strategy space which is required to be analyzed to find all the symmetric pure-strategy Nash Equilibria can be seen in Eq. (2.13).

\[
\{s, s', \ldots, s'\}, \forall s \in S, \text{ and } \forall s' \in S \quad (2.13)
\]

Thus, due to the symmetry of the game, the analysis is independent of the number of players.

2.6 Results

I run simulations to analyze the efficiency of the barter mechanism as I did in Section 2.3. The simulations were executed with the same parameters such that I can compare the barter based mechanism to the other two analyzed cases: 1) when the messages disseminate ideally (this case gives an upper bound for the goodput of the network) and 2) when the nodes download only primary messages.

As I have already described, the mobile nodes do not change their strategy during a game. Therefore, in each simulation run, the mobile nodes play a predefined strategy chosen from discrete values of the strategy space. The discrete values are the values from 0 to 1 increasing by 0.05.

I run a simulation with a concrete parameter set six times, and I consider the average goodput of player NULL. The obtained goodput of the other mobile nodes is irrelevant, because the game is analyzed from one, representative player’s point of view according to the description in Section 2.5.

Due to the above described discretization, each mobile node’s strategy can take 21 possible values. This means that I had to run \(21^2 = 441\) simulations for each parameter setting in order to find the pure strategy symmetric Nash Equilibria. The best response function of some parameter settings can be seen in Figures 2.7(a) and 2.7(b).
In Figure 2.7, on the vertical axis, there are the strategies that player NULL can choose, while on the horizontal axis, the strategy space of the other players is placed. The Nash Equilibrium candidates are the strategy profiles where player NULL and the other players choose the same strategy; these are denoted by solid, black points in Figure 2.7. Whereas, the best response strategy of player NULL to a specific strategy profile of the other players is denoted by empty circles. E.g. Figure 2.7(a) shows the result of a simulation set where the messages devaluate according to the function \( \delta \) (see Eq. (2.1)) the popularity of the generated messages is 0.4 and mobile nodes move according to the restricted random waypoint model. In this parameter set, the player NULL can get the highest payoff if its strategy is 0.15 independently from other player’s strategy. According to this, the Nash Equilibrium is the strategy set where all the nodes play with strategy 0.15. In other simulation sets the best response strategy value in the most cases is independent of the other players’ strategy, but the value of the best response is different. To give an overview of the value of player NULL’s best response in all simulation sets, I plotted a histogram in Figure 2.10 (investigated later on).

Figure 2.7: **Best response**: Nash Equilibrium candidates are denoted by solid, black points ●, while empty circles ○ show the best response strategy of player NULL. The Nash Equilibria are the strategy profiles where the best response function meets the Nash Equilibrium candidates.

In Figures 2.8(a) and 2.8(b), the results of simulations are plotted in an extended form. In these figures the payoff of player NULL is plotted against the strategies of player NULL and other players. The best response strategy of player NULL is the strategy where the payoff of the player NULL is maximal given a fixed strategy of the other players. The best response strategy is denoted by big black circles in Figure 2.8.

As one can see, the payoff of player NULL intensively falls down if player NULL does not cooperate \((s = 0)\). The nodes are encouraged to carry messages when the barter mechanism is used, because their goodput is higher if they do so (even if they are not directly interested in those messages). The payoff of player NULL intensively falls down too, if it is too altruistic \((s = 1)\), namely if it values the secondary messages as high as their primary messages. It helps the other mobile nodes, but it misses to obtain messages that it is interested in and suffer from goodput decrease.

To understand the reasons, I created some statistics during the simulations concerning the number and the type of message exchanges. In Figure 2.9(a), I plotted the number of all message exchanges against the strategies of all players. Note that the effect of a single node is negligible, this is why the player NULL and other players are not separately shown. I also classified the downloads by the type of the downloaded message (primary or secondary), these are plotted in
**2. Stimulating Cooperation Using Barter in Delay Tolerant Networks**

Figure 2.8: **Gain of player NULL**: The gain (goodput) of player NULL is plotted against the strategy of player NULL and other players. The best response function of player NULL is denoted by black circles.

Figure 2.9(b) and 2.9(c), respectively. The results shown there are related to the RRW mobility model.

The Figure 2.9(a) shows that the message exchange significantly decreases when the mobile nodes do not cooperate at all \((s = 0)\). As the message exchange decreases, the messages disseminate slower and the mobile nodes suffer from decreasing goodput.

However, the mobile nodes also reach lower goodput if they are too altruistic. The reason is the following: As one can see in Figure 2.9, when a player increases its secondary/primary value, the number of obtained primary messages decreases while the number of obtained secondary message increases, whereas the number of message exchange does not vary appreciably (not taking into account when the mobile nodes do not cooperate at all). This shows that the mobile nodes following an altruistic strategy do not utilize the investment of downloading secondary messages, but download more secondary ones.

To conclude the result of the simulations, I can state that in the simulated cases, the strategy which is most beneficial individually – the Nash Equilibrium of the barter game – to set the secondary/primary ratio to a low value but not to 0. Therefore, it is beneficial to help the other nodes \((s \neq 0)\) carrying their messages when the nodes exchange messages in a fair manner. However, if they are too altruistic, they download primary messages with less probability, and their goodput decreases. This can be seen in Figure 2.10, where the histogram of the Nash Equilibrium strategy values is plotted. The Nash Equilibrium values are obtained from all the simulation sets and grouped by the mobility models.

As one can read from Figure 2.10, there is no scenario where the most beneficial behavior is not to carry any secondary messages, because the S/P strategy values never equal to 0 in Nash Equilibria. The most preferred Nash Equilibria S/P value is 0.05, which is the lowest value in the considered simulation for collecting secondary messages. The highest secondary/primary ratio is 0.4, which means that even in the most special case a primary message worth much more than a secondary one. As a conclusion, the best strategy in general is to prioritize primary messages to secondary ones, but carry secondary messages too.

In Figures 2.11(a) and 2.11(b), the network goodput is plotted against the popularity attribute of the generated messages with restricted random waypoint and SUMO mobility model, respectively. It was done also in Figure 2.6, but these figures are supplemented with the goodput in Nash Equilibrium of the barter game. As it can be seen, the barter mechanism eliminated the selfish carrier effect by the principle of sharing a message with other party only if it can give a new message in return. In particular, the barter mechanism increases the goodput in the networks where the popularity value of generated messages is low. Furthermore, the goodput is close to the achievable goodput.
As one can see, as the popularity value increases, the goodput decreases in the case of barter mechanism. Recall that the goodput is the ratio of the obtained value and the maximal value of primary messages. Note that in each simulation the messages are generated at a fix rate, while increasing the popularity value means that a mobile node is interested in a larger subset of the messages. Therefore, the maximal value of primary messages increases. Meanwhile, the obtained value is limited due to the implicit cost, thus, the goodput decreases when the popularity value increases. The implicit cost is a system property, therefore it cannot be compensated. Note that in each simulation run, all the messages are generated with the same popularity value. The results that Figure 2.11 shows can not be utilized to derive how the barter mechanism handles messages with different popularity values in the same simulation. This is considered as a future work.

2.7 Future work

In this chapter, the idea of the barter has been arisen and investigated. I found many options to improve both the system model and the proposed mechanism. Even if an extensive simulation was run, the investigation can be extended in many ways in order to better understand how the barter mechanism perform in different scenarios.

The system model can be generalized in the following way:
Stimulating Cooperation Using Barter in Delay Tolerant Networks

Figure 2.10: Histogram of Nash Equilibrium values

- Currently, only the message nodes generate messages, which assumption came from the project for which the barter mechanism was proposed. However, in general DTNs, it is not the case, and the mobile nodes can generate messages too. This limitation should be relaxed. I think that it has no effect on the efficiency of the barter mechanism, however, it requires more investigation.

- In my system model, I assumed that the mobile nodes have unlimited memory. Considering the smart phones of today, practically this assumption is valid, however, the investigation can be more general if this assumption is relaxed. If I assume that the mobile nodes have limited memory capacity, the message exchange protocol can be modified in the following way: A mobile node downloads a message only if it has free space or it can obtain a message with higher value than the one which has the lowest value in its memory. This new approach also requires some investigation.

The message exchange protocol can be improved in the following ways:

- Currently, the mobile nodes download all the messages including the invaluable ones if they have connection, because of the simplicity of the message exchange protocol. Therefore, there
is a huge overhead. My main goal was to prove that barter mechanism stimulates cooperation, not to minimize the overhead. However, the overhead can be reduced by a simple solution: the mobile nodes download only messages which have higher value than a threshold. This approach is promising, because I run simulations with higher delete threshold values, and it shows that between 0.05 and 0.5 threshold values, the goodput does not change considerably, but increasing the delete threshold values results in fewer message exchanges. These results are not published here, because this investigation has not been completed.

- The mobile nodes maybe achieve higher goodput if they prioritize old messages — that will be erased from the network soon — to the fresh ones — which will be available later to but for less value.

Also the investigation can be improved in the following way:

- In the simulation runs, the messages have been generated with the same popularity values. This simplification does not make possible to investigate which is the most beneficial way of ordering the messages with different popularity values. One can think that if a message is more popular, it is worth to prioritize it, because the node can offer it for more nodes. One can think that it does not worth to prioritize the popular messages, because these are downloaded by more nodes anyway and maybe the other nodes think that it worth to download, therefore, the message become widespread, and no one wants to download anymore, in particular, its barter value decreases quicker than less popular messages.

2.8 Summary

In opportunistic networks, selfish nodes can exploit the services provided by other nodes by downloading messages that interest them, but refusing to store and distribute messages for the benefit of other nodes. To eliminate the harmful influence of selfish behavior, I proposed a mechanism which is based on the principles of barter. The users trade in messages, meaning that they can download a message from another user if they also provide a message in return. I analyzed my proposed solution using a game-theoretic framework, and showed that it indeed discourages selfishness. More precisely, the analysis shows that it is worth for users collecting, carrying and disseminating messages even if they are not interested in them, which has a positive effect on quality of data dissemination. In particular, the results show that, in realistic scenarios, the message delivery rate considerably increases if the mobile nodes follow the Nash Equilibrium strategy in the barter mechanism compared to the data dissemination protocol when no encouraging mechanism is present.
Chapter 3

Hide-and-Lie for enhancing privacy in Delay Tolerant Networks

3.1 Introduction

Without privacy protection, no new technology should spread widely. The privacy of the users must be ensured in DTNs as well. Some of the problems can be mitigated by traditional technologies, but some new problems are introduced by the store-carry-and-forward operation principle of the DTNs.

The privacy of a system and the anonymity of the users can be a problem on different layers of the communication stack. Using a rough partitioning, the problem can be related to the physical layer of the network, the network and transport layers, and the application layer. In the physical layer, the physical characteristics of the wireless transceiver, in the application layer, the application data, and in every layers the identifiers must be hidden or anonymized.

The main contributions of this chapter are the following: I am the first who raise the problem of application layer privacy in Delay Tolerant Networks. I characterize and simulate efficient software based attackers that can link different appearances of the same node with high probability using only regular handheld devices. I suggest and evaluate an efficient defense strategy, too, which is useful against this attacker without jeopardizing the node’s main goal, the message collection. All these contributions have been published in [Dóra and Holczer, 2010].

This chapter is organized as follows. Through giving an overview of the privacy related state-of-the-art in Section 3.2, I set the focus on the problem of the application layer privacy in DTNs. In Section 3.3, I describe the system model. The attacker model is presented and four different attackers are defined in Section 3.4. In Section 3.5, I describe my proposed privacy enhancing technique, called Hide-and-Lie Strategy. The simulation environment is defined in Section 3.6. The efficiency of different attackers and the privacy enhancing technique is exhaustively analyzed by means of simulations in Section 3.7. In Section 3.8, some further research directions are investigated. Finally, I sum up this chapter in Section 3.9.

3.2 State-of-the-art

The physical layer privacy problems are also referred to as remote device fingerprinting or remote device identification. In [Kohno et al., 2005], the authors can fingerprint a device remotely without any modification on the target machine, from any distance, measuring only the clock skew of the target machine. This technique can be very accurate, but needs very long interaction time. In [Franklin et al., 2006], a wireless device driver identification technique is given, which can reliably identify a device driver remotely, but cannot differentiate between devices using the same driver. In
3. HIDE-AND-LIE FOR ENHANCING PRIVACY IN DELAY TOLERANT NETWORKS

[Brik et al., 2008; Čapkun et al., 2007], very accurate radio fingerprint based device identification methods are given which can identify the devices very precisely, but it utilizes special expensive hardware equipment.

In the network and transport layer, a similar problem to the DTN’s arises in Vehicular Ad-hoc Networks (VANET). VANETs are similar to DTNs in many aspects. They are mobile, the transaction times are short, and each device belongs to a well defined person. However, in VANETs, some infrastructure elements can be assumed in contrast to DTNs. Nevertheless, the solutions proposed for privacy issues can be a good inspiration for solving the privacy problems in DTNs. Many authors addressed the privacy problem in VANETs, (e.g., in [Dötzer, 2005; Hubaux et al., 2004; Raya and Hubaux, 2007]). An overview of the problem of providing location privacy for VANETs is given in [Gerlach, 2006]. This paper shows that messages contain state dependent information (e.g. speed, location, time) and contain no personal information.

The problem of privacy in opportunistic networks is considered in some papers. In [Lilien et al., 2006], the authors raise the problem of data privacy when a node sends sensitive information to another node and it does not want it to be available for intermediate nodes. The author of [Heinemann, 2007] proposes a pseudonym generating technique using public keys for supporting anonymous communication. In [Kate et al., 2007], an anonymous communication solution is presented and a new anonymous authentication protocol is introduced for DTNs. The solution is based on identity-based cryptography and solves the problem of anonymous communication on the network layer, but the application layer problems are not handled.

Privacy preserving data mining is also a relevant topic. In [Verykios et al., 2004] the categorization of different privacy preserving techniques is given. The problem of deriving private information from randomized data is analyzed in [Huang et al., 2005]. In [Du and Zhan, 2003] the authors investigate how to randomize some data, while keeping the statistics close to the original. In DTN, a set of nodes can be viewed as a distributed database and I want to preserve the privacy of each record. From this point of view, these techniques are very promising, but from other point of view, the proposed methods in data mining sometimes use techniques which are not suitable for moving nodes with low bandwidth.

It is essential that the communication be anonymous, otherwise an attacker can trace a user by following the appearance of its identity. Anonymity (or at least pseudonymity) can be easily achieved at the network layer by the usage of pseudonyms (i.e., temporal identifiers). A more serious and DTN specific privacy problem is that the users can be identified by the messages stored in their device. If an attacker is able to build a user profile using the exchanged application data, the user becomes traceable even if the communication is completely anonymous. Therefore, a new mechanism or an adaptation of some proposed mechanism is required in DTNs to ensure untraceability of the nodes, namely, to avoid building traceable user profiles.

3.3 System model

The system model highly relies on the one described in Chapter 2. Nevertheless, for easier readability I repeat the common parts and emphasize the properties characteristic to the privacy problem, here.

In my model, the users are placed on a field of arbitrary two-dimensional shape. They own devices which can communicate with each other within their radio range. The used wireless technology can be Bluetooth, Wi-Fi, or any suitable wireless technique. The messages are generated and disseminated among the devices/users but each user is only interested in a subset of messages. The dissemination process is based on the store-carry-and-forward principle. A node includes a user (the owner of the device) and a device. I assume that the data dissemination has no impact on the user’s movement.

The communication between the nodes is assumed to be anonymous. Hence, an attacker is not able to trace a node by e.g. simply tracing a network identifier.

The messages are generated by special nodes, called message generator nodes. In my system model, the time is slotted, and each message generator node generates a new message with a fixed
average rate: \( \rho \) messages per time step. The message generator nodes are static and each one stores only the most recently generated message. This message can be downloaded by any node that passes by the message generator node.

I assume that a mechanism can filter out the fake messages from the network. This assumption is necessary, otherwise, an attacker is able to create a special decoy message and trace a node by following it. The analysis of the effectiveness of this kind of attacks and countermeasures is out of the scope of this thesis. However, to justify that this assumption is realistic, I introduce some mechanisms:

1. A node downloads only those messages that are generated by a trusted entity and the authenticity is proven by a digital signature.
2. A node downloads any message but offers only those whose reputation is higher than a threshold. The reputation can be computed automatically or manually based on the content of the message.
3. A node downloads any messages but offers to other nodes only the widespread disseminated messages. This technique can be used only by the minority of the nodes.

For the sake of simplicity, it is assumed that there are \( C \) categories, and each message belongs to a single category. When a message generator node generates a message, it specifies which category the new message belongs to. Each message is classified into a category uniformly at random. Therefore, a new message belongs to a specific category with probability \( \frac{1}{C} \).

An interest profile \( (IP) \) of a node, which is a part of the user profile, is a binary vector representing a list of categories the node is interested in. A message belonging to category \( k \) is called primary for a node if \( IP[k] = 1 \), the message is secondary otherwise. A node is interested in any given category (i.e. all the messages belonging to that category are primary for the node) with probability \( \varepsilon \). As the participating nodes are interested in at least one category, the case when \( \varepsilon = 0 \) can be excluded, therefore, the cases when \( 0 < \varepsilon \leq 1 \) are considered. For the sake of simplicity, \( \varepsilon \) is equal for each node in each considered scenarios.

Each message is assumed to have a unique identifier and a node can decide based on this identifier if a message \( M \) is stored in its memory before downloading from another node.

According to my assumptions, a message \( M \) has the following formula:

\[
M = [ID|CAT|data]
\]  

where the \( ID \) is the unique identifier of the message, \( CAT \) is the identifier of the category which the message belongs to, and \( data \) is the content of the message. The length of the \( data \) may be some magnitude higher than the length of the \( ID \) and the \( CAT \).

When two nodes get in the vicinity of each other, they start to exchange messages. Each node \( u \) wants to download those messages from the other participant that \( u \) does not store and fit to its interest profile.

The whole message exchange may be not completed because the nodes are mobile and they may leave the radio range of the other participant before exchanging all the required messages. Therefore, according to the system model, the nodes are not able to obtain as many messages as they want but at maximum one for each participant per time step. It is assumed that a message is downloaded without interruption in a time step.

In my imagined scenario, the batteries of the handheld devices can be easily recharged day by day, hence, the cost related to the battery consumption due to communications is negligible. However, the average number of downloaded messages per node is investigated in the simulations to get an insight of the message exchange rate.

The storage cost has two aspects: 1) The messages need storage space and storage constraints may limit the number of stored messages. No explicit limitation for the storage space is defined, however, the maximum quantity of the stored messages in the devices is investigated. 2) The time needed to determine which messages the nodes want to download increases polynomially with the number of messages stored by the other participant. Therefore, the increasing number of messages...
3. HIDE-AND-LIE FOR ENHANCING PRIVACY IN DELAY TOLERANT NETWORKS

is controlled by deleting the messages of the system. Each message is deleted \( D \) time steps after its generation. \( D \) should be sufficiently large, as the network itself is delay tolerant.

In order to investigate the effect of the defense mechanism on the message delivery ratio, a formula is defined for the gain of the nodes. Until time step \( t \), a node \( u \) obtains \( O_u(t) \) number of primary messages while in the system, there have been \( A_u(t) \) number of messages generated which is primary for \( u \). The gain \( (G_u(t)) \) of node \( u \) is the ratio of these values.

\[
G_u(t) = \frac{O_u(t)}{A_u(t)}
\]  

(3.2)

The method shown in Section 2.3.2 can be easily adapted to prove that the gain approaches its steady state value in time. Therefore, in what follows, I approximate the steady state value of the gain by considering the gain after sufficiently large simulation time and I denote it by \( G_u \).

From the privacy point of view, an extreme approach consists in denying the participation in the network. I assume that the nodes want to obtain messages while they want to preserve their privacy, too. A beneficial and in the same time selfish behavior would be to download any kind of messages but not forwarding them. To prevent this kind of selfish behavior, I assume that a mechanism encourages the message dissemination among the nodes. This could be electronic payment, reputation based or any alternative solution such as shown in Chapter 2. Therefore, the nodes themselves want to increase the number of offered messages. Note that such a mechanism may affect the attackers behavior. For the sake of simplicity, this effect is eliminated from the investigation and it is considered as a future work.

3.4 Attacker model

The attacker wants to track the target node to breach its privacy. To do so, it tries to link the profiles acquired in different times together. If the profiles can be linked correctly, the attack against the privacy is successful.

Note that a protection against traceability also provides protection against revealing the nodes’ true interest profile. If an attacker was able to profile a node, it could also trace her. Furthermore, for untraceability, it must be assured that an attacker cannot determine a consistent, but maybe fake interest profile.

In this section, I describe what information an attacker can get from the nodes, how he can obtain this information and how he can link the nodes.

3.4.1 Leaking information

The communication between the nodes can leak some information about the interests of the participants. In this chapter, attacks based on these leaked information are considered. I assume that the attacker can estimate the following user profile \( (UP) \) from a node \( u \) at time \( t \):

\[
UP_u(t) = (EIP_u(t), CHM_u(t), IDL_u(t))
\]  

(3.3)

The \( UP \) consists of the following triple:

- Estimated Interest Profile \( (EIP) \) is a binary vector. The value of the vector at the \( k^{th} \) position equals to 1 if category \( k \) seems to be interesting for node \( u \).
- Category Histogram of offered Messages \( (CHM) \) shows, for each category, how many messages in the ID list belong to that category.
- \( IDL \) is the ID list of offered messages.

In this document, I abstract away what message exchange protocol is used, I only assume that an attacker can obtain the current value of the \( UP_u(t) \) for each node \( u \) in time step \( t \). However in the rest of this section, I show how an attacker can obtain the \( UP_u(t) \) by participating in the
message exchange protocol. Even though there are many possible message exchange protocols, they can be classified into two groups: push and pull-based mechanisms. I define one mechanism for each and I show how an attacker can obtain these triples.

**Push-based message exchange protocol** When two nodes get in the vicinity of each other, they interact as it is shown in Figure 3.1(a). First, node $A$, which starts the communication, sends a list of the stored messages ($L_A$) consisting of the ID and the category $CAT$ of each message. $B$ sends back a list of required messages ($L_{AB}$) containing the IDs of those ones which are primary for $B$ and $B$ does not store. $A$ sends the content ($D_{AB}$) of each message listed in $L_{AB}$. In the second part of the protocol, the roles change and $A$ obtains through the same steps those messages which are primary and not stored in its memory.

An attacker can obtain the triple with the following mechanism. Considering $B$ as an attacker, it can easily calculate the CHM and the IDL by obtaining $L_A$. In the second part of the message exchange, $B$ creates a special $L_B$. $B$ first collects those categories ($C_A$) which were not present in $L_A$. Then, $B$ creates $L_B$ such that each category from the list $C_A$ is represented at least by one message. Getting the response $L_{BA}$, $B$ reads what are the categories that $A$ is interested in but it could not obtain or deleted the messages belonging to those categories before. The EIP can be calculated by getting the union of the categories of the stored messages ($L_A$) and the category of the required messages ($L_{BA}$).

![Diagram of message exchange protocols](image)

Figure 3.1: Message exchange protocols

**Pull-based message exchange protocol** When two nodes get in the vicinity of each other, they interact as it is shown in Figure 3.1(b). First, node $A$, which starts the communication, sends a list of categories ($P_A$) according to its IP. Node $B$ collects the ID of messages belonging to categories listed in $P_A$ ($L_{BA}$) and sends to $A$. $A$ removes from $L_{BA}$ the IDs which is already stored and sends back the list ($L'_{BA}$). $B$ sends the contents of the required messages ($D_{BA}$). In the second part of the protocol, the roles change and $B$ obtains through the same steps those messages which are primary and not stored in its memory.

An attacker can get the same set of information as in the push-based message exchange protocol. Considering $B$ as an attacker again, $B$ can easily get the EIP from the $P_A$. In the second part, $B$ can claim that it is interested in all the messages and list all categories in $P_B$. As a response, $A$
will send the list of stored messages, and $B$ can get $CHM$ and $IDL$ as I have shown in the push model.

### 3.4.2 Attacker behavior

The attacker, in my model, behaves according to the following attacker model:

1. The attacker identifies its target node ($u_T$) from $N$ nodes.
2. The attacker reads the current user profile of the target: $UP_{u_T}(t_0)$. The time step when this happens is considered as a reference time, i.e. $t_0$.
3. $\tau$ time later ($t_1 = t_0 + \tau$), the attacker reads $UP_{u_i}(t_1), i \in [1..N]$ of each node and calculates a metric how similar is $u_i$ to $u_T$. $\tau$ is referred as the attacker delay. In order to mislead the attacker, the nodes can slightly modify their $UP$s. The $UP$ perturbation is defined in Section 3.5.
4. The attacker chooses the node most similar to the target node. If more than one have the maximal similarity value, it chooses randomly between them. If the chosen node is $u_T$, the attacker is successful.

I have chosen for the analysis the success probability of the attacker as the privacy metric, because it is widely used and tells the most about the expected outcome of the attack. In the cryptographic literature, a widely used metric is the indistinguishability of the target from a randomly chosen node [Menezes et al., 1996]. This metric differs from ours slightly as the attacker wants to distinguish the target from every other node. My extended metric can be imagined as the conventional metric used $N$ times one after the other. More precisely, if the attacker can recognize its target from two nodes with probability $p$, then it can recognize it from $N$ nodes with probability $p^{N-1}$, if the nodes are independent. The conventional model is more sensitive for $p$ close to 0.

In contrast, the extended model is more informative for $p$ close to 1. As the results show, $p$ can be close to 1 when no defense mechanism is used, so the extended model is used.

To fully define the attacker, a similarity metric must be defined. Some possible and useful similarity functions are defined in the next section.

### 3.4.3 Attacker functions

The attacker can define the similarity of the target and a suspected node based on the $UP_u(t)$. Using the user profiles of the nodes, the attacker can calculate the similarity using an attacker function $A$.

More formally the input of $A$ are $N + 1$ user profiles, and the output is an ID of a node:

$$A : (UP_{u_T}(t_0), UP_{u_i}(t_1), i \in [1..N]) \rightarrow j, j \in [1..N] \quad (3.4)$$

The attack is successful if and only if $j = T$.

It is clear that any attacker can reach a minimal value of the success probability $\frac{1}{N}$ by simple guessing. Higher values can also be achieved using more sophisticated attacker functions. In the following, four different simple attacker functions are defined.

**Prefiltered ID Based attacker function** assumes that nodes show their real interest profiles. The attacker can filter out every suspect who has different $EIP$s, considering only the nodes whose $EIP_{u_i}(t_1)$ equals to $EIP_{u_T}(t_0)$. From the remaining set, it selects the one whose $IDL_{u_i}(t_1)$ is the most similar to $IDL_{u_T}(t_0)$. Under similarity, the cardinality of the intersection of the target ID list and the suspect’s ID list is meant. If the remaining set is empty, the attacker selects the target by pure guessing. The intuition behind this attacker is that after some time the target can get some new messages and delete some old ones, but mainly its ID list is unchanged. This attacker can be very efficient if the nodes show their real $IP$s which means that $EIP$s are not changed over time, but can be very inefficient if the $EIP$s are changed.
Unfiltered ID Based attacker function is a simplified version of the previous function, as it uses only the cardinality of the intersection of $IDL_{u_2}(t_0)$ and $IDL_{u_1}(t_1)$, but it does not prefilter the nodes by their $EIP$. This attacker is not so efficient in case of time invariant $EIP$s, but less sensitive for changing $EIP$s.

Category Histogram Based attacker function selects the node $u$ whose $CHM_u(t_1)$ is the most similar to the $CHM_{u_2}(t_0)$. The similarity of two histograms is calculated using the $\chi^2$-test. The intuition behind this attacker function is that a node can show a modified $EIP$ but the histogram represents its real interest profile if the node collects messages according to its real interests.

Significant Category Based attacker function is the most complex function analyzed in this chapter. It assumes that the interested categories are overrepresented in the ID list and the uninterested categories are underrepresented. This categorization only depends on the real $IP$ of the target, and is hard to influence without totally changing the $IP$. To find the interested categories, the $C$ categories must be classified into two clusters: the significant categories, and the remaining categories. This task can be easily done using the k-means clustering algorithm [Hartigan, 1975] on the $CHMs$. The result of the clustering is a binary vector of length $C$ with ones at the significant categories. The similarity of two binary vectors is defined as the Hamming distance of the vectors.

The properties and efficiency of the different attacker functions are analyzed in Section 3.7.

3.5 My approach

In order to preserve a node’s privacy, the User Profile ($UP$) should be obfuscated to deceive the attacker. Against an eavesdropping attacker, another solution is to design the message exchange protocol in a way that it ensures that no sensitive information can leak during the communication. I am focusing on developing an obfuscation based mechanism because that can be used if the attacker actively takes part in the communication.

The more $UP_u(t_1)$ is different from $UP_u(t_0)$, the less likely the attacker can link the two profiles. The continuously changing profile hardens the task of the attacker, however, it may thwart the node from collecting primary messages. This is thoroughly analyzed in Section 3.7.

Two simple methods can be used to modify the $UP$ through modifying the Interest Profile ($IP$) of the node. The first one is to hide some interesting categories, and claim them as uninteresting. The second one is to lie about some uninteresting categories, and claim them as interesting. These techniques can be used at the same time, this is what I call Hide-and-Lie Strategy (HLS). The temporarily obfuscated $IP$ is the Temporal Interest Profile or the $EIP$ from the attacker point of view. The $EIP$ can be transient, which means that a new $EIP$ can be generated by every node in every time step.

Obviously, the required and offered messages during the message exchange must be synchronized with the $EIP$: 1) messages relating to hidden categories must be hidden as well and no message of hidden category should appear on request list, and 2) when a node lies about being interested in a given category, it collects and offers messages belonging to that uninteresting category.

Many different HLSs can be envisioned. Different strategies can hide or lie about different categories in different situations. In the following, a simple but rather general solution is given: every node generates its $EIP$ from its $IP$ by inverting every category in the $IP$ with a given probability $\lambda$. Inverting means indicating an uninteresting category as interesting or vice versa. This parameter $\lambda$ is the Hide-and-Lie strategy value.

As $\lambda$ is a probability, it is between 0 and 1. The nodes which do not use any obfuscation techniques can be modeled as nodes using HLS with $\lambda = 0$ as the node never modifies its $EIP$.

The other interesting value of $\lambda$ is 0.5. It totally randomizes the $IP$, making the $EIP$ a uniformly distributed random binary vector. It is the best strategy for the privacy sensitive users. For demonstrating this, let us assume two nodes, $u_1$ and $u_2$ at time $\tau$ of the attack. They show temporal interests in every category with probability 0.5, thus, their interest profiles are independent from their real interest profiles. If $\tau$ is greater than the $\ell$ message expiration time, then none
of them has any messages from the reference time of the attack \( (t_0) \). On average, every user shows interest in a given category in every second round (as \( \lambda = 0.5 \)), so every message is collected by every node with the same probability. The CHM of the nodes are close to the uniform distribution considering those categories where the EIP shows that the node is interested in (other categories are represented by 0 messages). The reason is that every message is generated and collected with the same probability. Therefore, \( u_1 \) and \( u_2 \) show user profiles, which are independent from the user and statistically the same.

The used HLS transformation of the users’ profiles generates every possible UP with the same probability, thus, no statistical test can distinguish between \( u_1 \)’s and \( u_2 \)’s UP.

Values of \( \lambda \) greater than 0.5 are useless for the nodes, as they make the EIP as traceable as the inverse EIP with \( \lambda' = 1 - \lambda \), but the nodes collect more uninteresting messages than interesting ones. Consequently, in the following, only \( 0 \leq \lambda \leq 0.5 \) are considered.

### 3.6 Simulations

In the simulations implemented in C++, the fixed-number of mobile nodes move in discrete time steps according to one of the two mobilities: the random walk (RW) and restricted random waypoint (RRW) model. The simulator is based on the one described in Chapter 2, however, I repeat the common parts for the sake of better understanding.

In the RW model, 300 nodes move on a grid of size 15 \( \times \) 15. In each time step, a node can move to one of the four neighboring grid points (in what follows, these are called meeting points), or stay at the current place. The probability of each of these actions is 0.2. In each time step, the nodes that happen to be at the same meeting point are paired randomly and each pair executes the message exchange protocol. These pairs are able to download one message from each other as described in Section 3.3.

In the RRW model, 300 nodes (initially placed uniformly at random) move on a field of size 20 \( \times \) 20 unit. On the field, there are some special points chosen at random; these are called meeting points. Each node selects a meeting point randomly, and moves towards this meeting point along a straight line with a fixed speed. When the meeting point is reached, the mobile node stops and stays there for randomly chosen time (10 time steps on average). Then, it chooses another meeting point and begins to move again. The nodes that happen to be at the same meeting point in the same time step are paired randomly and these pairs are able to download one message from each other as described in Section 3.3.

In the case of RW, 30 message generator nodes are placed on the subset of all meeting points uniformly at random. In the case of RRW, one message generator node is placed on each meeting point, and the number of meeting points is 30. All the 30 message generator nodes together generate one new message per time step on average both in case of RW and RRW mobility model.

The parameters of the mobility models were determined such that the number of message exchanges are equal on average.

The length (number of time steps) of the simulation was determined in an empirical way by taking into account that the gain has to reach its steady-state value. In the beginning of the simulation, the nodes do not store any messages. Therefore, their gains are volatile in the first time steps. When the simulations were run for 3000 time steps, the average gain has not changed considerably for upcoming 1000 time steps in the analyzed simulations. Therefore, the attacker started its attack after 3000 time step long bootstrap \( (t_0 = 3000) \) and the simulator was run for additional 1000 upcoming time steps to investigate the effectiveness of the attacker for different \( \tau \) values.

Some of the parameters that describe my envisioned system were fixed in order to reduce the number of simulation scenarios and other ones which have the highest effect on the success probability of the attacker were varied. The fixed simulation parameters are summarized including the mobility model specific ones in Table 3.1.

The simulation parameters that are related to the interest profile were varied: number of
### 3. HIDE-AND-LIE FOR ENHANCING PRIVACY IN DELAY TOLERANT NETWORKS

<table>
<thead>
<tr>
<th>Table 3.1: Fixed simulation parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>Simulation length in time steps</td>
</tr>
<tr>
<td>Number of nodes (N)</td>
</tr>
<tr>
<td>Number of message generator nodes</td>
</tr>
<tr>
<td>Message generation rate (¿)</td>
</tr>
<tr>
<td>Simulation area (unit)</td>
</tr>
<tr>
<td>Number of meeting points</td>
</tr>
<tr>
<td>Probability of leaving a meeting point</td>
</tr>
<tr>
<td>Velocity</td>
</tr>
<tr>
<td>Lifetime of messages (¿)</td>
</tr>
</tbody>
</table>

message categories (¿) and probability of being interested in a category (¿) as these parameters affect most the success probability of the attack.

Parameter C should be higher than 1, otherwise all the nodes have the same IP. Therefore, the chosen lowest value is 2. I think that 50 categories is high enough, because a higher value would not affect the simulation results considerably (the results for 30 and 50 are similar). The considered values are 2, 5, 10, 30, 50.

To reduce the complexity of the simulations, I selected some 0 < ¿ ≤ 1 values in a way that instead of ¿ = 0 I included ¿ = 0.05, and I also investigated a special case, ¿ = 0.5. The considered values are 0.05, 0.2, 0.4, 0.5, 0.6, 0.8, 1. Note that with probability (1−¿)C, the simulator generates such an interest profile that the node is not interested in any category. In that case, a new IP is generated.

Recall that a node can choose a Hide-and-Lie strategy value from the interval 0 ≤ λ ≤ 0.5. For the sake of simplicity, only those cases are considered when each node chooses the same λ value from the following set: {0, 0.1, 0.2, 0.3, 0.4, 0.5}.

I also investigate how the time elapsed between t₀ and t₁ (i.e., τ) affects the success probability of the attack. I considered the following values of τ: {1, 50, 250, 500, 1000}. In the special case when τ = 1, the ID list of stored messages does not change considerably, however, the Hide-and-Lie Strategy affects the UP. Recall that the messages are deleted from the system after 500 time steps. Therefore, when τ > 500, no message will match to the target node’s ID list in t₀.

<table>
<thead>
<tr>
<th>Table 3.2: Varied simulation parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>Number of categories (¿)</td>
</tr>
<tr>
<td>Probability that a node is interested in a category (¿)</td>
</tr>
<tr>
<td>Hide-and-Lie strategy value (λ)</td>
</tr>
<tr>
<td>Attacker delay (τ)</td>
</tr>
</tbody>
</table>

The main objectives are to investigate the success probability of the attacks and the efficiency of the HLS. The analysis was performed in each combination of the parameter values summed up in Table 3.2. In one parameter set, the success probability of every attacker function was calculated using the following method: Only one simulation run for each parameter set was executed. In each execution, the attacker selects each node as a target node one-by-one at time t₀ and performs the attacker function with the target node and all the nodes as the input of the function at each t₁ = t₀ + τ time. The success probability is the ratio of the successful attacks.
3.7 Results

In this section, two representative scenarios (see Table 3.3) are exhaustively analyzed. In particular, the efficiency of different attacker functions presented in Section 3.4.3 and the efficiency of the defense mechanism presented in Section 3.5 are investigated. Beyond the analysis of two emphasized scenarios, I show the differences compared to the other simulated scenarios. In the two considered scenarios, I investigate the effect of the Hide-and-Lie Strategy on the reached gain and the number of downloaded primary and secondary messages and the maximum memory required to follow the proposed Hide-and-Lie strategy.

As my experience showed that the chosen mobility model does not affect the results considerably, I have selected the random walk mobility model in the analyzed scenarios. Because of the space limits, I emphasize rather the effect of the probability of being interested in a category instead of the number of categories. Therefore, in the presented simulation results, the number of categories is fixed to 30, which can be a realistic value for a lot of applications. In the two investigated scenarios, the probability of being interested in a category takes the values 0.05 and 0.4. The former value refers to those scenarios where the nodes are interested in a small subset of messages, while in the latter scenario, the nodes are interested in a large subset of messages. In Table 3.3, I summarize the parameter values of the scenarios beyond the already fixed parameters introduced in Table 3.1.

<table>
<thead>
<tr>
<th>Scenario 1</th>
<th>Scenario 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobility model</td>
<td>RW</td>
</tr>
<tr>
<td>Scenario 1</td>
<td>RW</td>
</tr>
<tr>
<td>Scenario 2</td>
<td>RW</td>
</tr>
</tbody>
</table>

The success probability of the attacker functions is plotted against different Hide-and-Lie strategy values (λ) and different attacker delay (τ) values of Scenario 1 and 2 in Figure 3.2(a) and 3.2(b), respectively. For the sake of better understanding, the plots are separated by different attacker delay values.

The Prefiltered ID Based attacker function assumes that the nodes do not apply any privacy enhancing technique. According to this, it is the most efficient attacker function when λ = 0, but in any other cases, the attacker function cannot distinguish the target node from the others, because even one entry changing in the EIP misleads the attacker.

A more robust solution can be obtained by omitting the prefiltering which results in the Unfiltered ID Based attacker function. The success probability of this function decreases when λ = 0 compared to the prefiltered function but considerably increases in other cases. The reason is that the number of all the combinations of the messages give enough variety to the attacker to identify the nodes with higher probability even if they hide a small subset of the messages when they meet other nodes. As the nodes increase the λ value, they collect messages from larger sets and they can hide more messages. Hence, the nodes are able to deceive the attacker with high probability. Therefore, the success probability of the attacker function decreases with the increasing λ value. If λ = 0.5, the attacker function is as inefficient as a naïve attacker.

The Unfiltered ID Based attacker function is very sensitive for the attacker delay. As τ increases the nodes delete more and more messages making the attack less and less efficient. Finally, the nodes delete all the messages that could match the IDL_{A,T}(t_0) after ℓ time steps and this attack becomes inefficient in cases where τ = 500 or τ = 1000. Recall that ℓ = 500 in the considered scenarios.

The Category Histogram Based attacker function is less sensitive to the τ value, but it is less efficient when τ is lower than the ID Based attacker function. The inefficiency of this attacker function comes from the fact that the Hide-and-Lie Strategy causes intolerable differences for the χ²-test when all the messages appear or disappear belonging to a category when EIP changes.

The attack that is least sensitive to τ is the Significant Category Based attacker function. The advantageous characteristic comes from the fact that this function tries to reveal the real interest
Figure 3.2: Success probability of $A$ as a function of the Hide-and-Lie strategy values ($\lambda$)
profile. However, it still does not work when the nodes hide their identity with \( \lambda = 0 \) strategy, because there are no over- and underrepresented categories in that case.

The Significant Category Based attacker function is the most efficient attacker function in Scenario 2, but it is less efficient in Scenario 1.

Taking all the considered attacker functions into consideration, I can conclude that the efficiency of the attacker functions changes according to the parameters of the model. However, a common tendency is that if the nodes apply the Hide-and-Lie Strategy with high value of \( \lambda \), none of the attackers is able to distinguish them better, independently of the value of \( \tau \), than a naïve attacker which picks up one of the nodes by random.

Even if an attacker can distinguish two nodes if their IPs are different (I call this attacker ideal IP based attacker \( A_{IP_{ideal}} \)), the probability that two nodes have the same IP is not negligible.

The success probability of an ideal IP based attacker can be viewed as an upper bound for any other IP based attacker, such as, e.g. the Significant Category Based attacker function. This value can be determined analytically. Through this analysis, I show how different \( C \) and \( \epsilon \) values affect the success probability of the attackers.

The success probability of the ideal IP based attacker is determined by the number of equal IPs. To compute the success probability, first the probability \( p \) of two IPs being equal is computed as follows:

\[
p = \sum_{w=1}^{C} \binom{C}{w} \left( \epsilon^2 \right)^w \left( (1-\epsilon)^2 \right)^{C-w} \frac{(1-(1-\epsilon)^C)^2}{(1-(1-\epsilon)^{C})^2} \tag{3.5}
\]

where \( w \) is the weight of the IP varying between 1 and \( C \) (recall that every node is interested at least in one category).

The success probability of \( A_{IP_{ideal}} \) is the reciprocal of the average number of nodes with the same IP:

\[
Pr(A_{IP_{ideal}}(UP_{u_T}(t_0), UP_{u_1}(t_1), \ldots, UP_{u_N}(t_1)) = u_T) \approx \frac{1}{1 + p(N-1)} \tag{3.6}
\]

The ideal values according to Eq. (3.6) are 0.341 and 1 for Scenario 1 and 2, respectively. These values are valid only for \( \lambda = 0 \), and confirmed by Figure 3.2. These values are shown in Figure 3.3, too, where Eq. (3.6) is plotted against different \( C \) and \( \epsilon \) values.

The characteristic of the success probability of the attacker in the case of the two emphasized scenarios are similar to each other as Figures 3.2(a) and 3.2(b) show and these are similar to the other scenarios which are simulated but not presented here. However, as Figure 3.3 shows, the success probability of the ideal IP based attacker depends on the parameter value of the number of categories and the probability of a node being interested in a category. As one can read from the figure, when there are large number of categories in the system, the success probability of an ideal attacker is high. On the other hand, when the number of the categories is low, the success probability highly depends on the value of \( \epsilon \). As the value \( \epsilon \) gets closer to 0.5, the success probability increases. The reason is that an attacker can distinguish nodes when the probability that the IPs of two nodes are equal is low. All these statements are confirmed by the simulation results that are not presented here, and these effects can be observed even in cases when \( \lambda > 0 \).

In Figure 3.4, I show the average gain of all the nodes as a function of the Hide-and-Lie strategy in the two scenarios and its empirical standard deviation. I have to stress that these two figures do not represent all the appeared characteristic of the figures, however, Figure 3.4(a) shows an interesting property of the Hide-and-Lie Strategy. Namely, increasing \( \lambda \) does not degrade but increases the data delivery ratio in some scenarios.

The Hide-and-Lie Strategy has two contradictory effects: On the one hand, when the nodes happen to hide what they are interested in, they may miss some primary messages to download.
3. HIDE-AND-LIE FOR ENHANCING PRIVACY IN DELAY TOLERANT NETWORKS

Figure 3.3: Analytically determined upper bound for success probability of ideal IP based attacker functions when 300 nodes are present in the network

![Success probability graph](image)

Figure 3.4: Average gain with the empirical standard deviation

(a) Scenario 1

(b) Scenario 2

On the other hand, when the nodes happen to lie being interested in some category, they store-carry-and-forward secondary messages, which increases the data delivery ratio in general [Buttyán et al., 2010a]. The cumulative effect depends on the system parameters. E.g. in a case when nodes are interested only in a small subset of categories and they do not carry secondary messages, they can exchange messages only with small probability. Therefore, the Hide-and-Lie Strategy in some cases can be viewed as a motivation to store-carry-and-forward secondary messages as it can be seen in Figure 3.4(a). On the other hand, when the nodes have many possibilities to get primary messages, the latter effect has no considerable benefit while the former effect degrades the gain. Surprisingly, the two effects are balanced in Scenario 2 as one can see in Figure 3.4(b).

Even though I did not take into consideration the energy consumption and the memory costs of the communication when I calculated the gain, I collected related information during the simulation. I plotted the average number of primary and secondary messages downloaded by one node and maximum memory usage as a function of the Hide-and-Lie strategy in the two considered scenarios in Figure 3.5.

As one can expect, the number of the downloaded secondary messages increases with increasing $\lambda$ value. The number of the downloaded primary messages changes as the gain changes because the gain is a normalized value of the number of obtained primary messages.
Figure 3.5: **Costs**: Average number of primary and secondary messages downloaded by a node and the maximum memory usage

Even though the gains are comparable in the two scenarios as Figure 3.4 shows, there is almost one order of magnitude difference in the number of downloaded primary messages. The reason is that in Scenario 1, the nodes are interested in 5% of the messages and in Scenario 2, the nodes are interested in 40% of the messages while the number of the generated messages does not change considerably in the two scenarios. Due to the same reason, the number of the secondary messages for a node is less in Scenario 2 than in Scenario 1. The ratio of the number of downloaded primary and the number of secondary messages is $\varepsilon(1 - \lambda) : (1 - \varepsilon)\lambda$.

Note that even if the nodes download more and more secondary messages as $\lambda$ increases, the maximal memory usage does not increase at the same order. Thus, the nodes do not need to maintain much larger memories when they want to protect their privacy.

### 3.8 Future work

In this chapter, the so-called Hide-and-Lie mechanism has been proposed and investigated both with extensive simulations and with analytical tools. There are many options to improve the protection, reduce the overhead, and better understand how Hide-and-Lie perform in real scenarios.

- Currently, the investigation of the Hide-and-Lie mechanism mainly rely on extensive simulations, however, the system and attacker model was designed such that it could be done (maybe with some further simplification) by analytical tools as well.

- In the current system model, each mobile node is interested in a category with some probability and these are independent of each other. However, in a real application this is usually not the case. E.g., those who are interested in the literature are more likely interested in art of painting than those who are not interested in the literature. The correlated categories is a future direction of improving the current solution.

- As mentioned above, an attacker may inject some decoy messages to trace a node based on the fact where the message appears. In order to be able to protect against such attacks a novel approach is required.

- Similarly to the previous chapter, the message nodes are the only ones who can generate messages. This assumption could be relaxed without any major consequences.

- One can find that increasing S/P also provides some randomization and obfuscation of own interests. The idea of combining the two mechanisms for cooperative and privacy enhanced
data dissemination make sense. However, it is far from trivial to implement. Barter mechanism ensures that it is worth to download secondary messages, but primary ones are still preferred. In contrast to this, the Significant Category Attacker can reveal over- and under-represented categories (can distinguish between primary and secondary messages).

- Also in a real application maybe the categories are not known a priori. Therefore, the nodes can carry messages for their "friends" in order to obfuscate their identity with valid messages.

3.9 Summary

In this chapter, the problem of application layer privacy in Delay Tolerant Networks has been investigated. In particular, an attacker can build a user profile of a node based on what messages the node stores and what messages it wants to download. After profiling, the attacker can trace the node based on the user profile even if the node communicates with the other nodes through anonymous links. A system and an attacker model was built and some attacker functions were proposed. A defense mechanism called Hide-and-Lie Strategy against such attacks was proposed, too. This mechanism has a free parameter with which the system can be tuned between high privacy level and low data-forwarding overload. In my model, I analyzed the efficiency both of the attacks at different parameter values and the proposed defense mechanism. I showed that without any defense mechanism, the nodes are traceable, but with the proposed Hide-and-Lie Strategy, the success probability of an attacker can be decreased substantially. The message delivery ratio and the costs at different Hide-and-Lie parameter values are also investigated. I found that in some scenarios, the Hide-and-Lie Strategy can be viewed as a motivation for other nodes to carry messages that they are not interested in. Therefore, as a positive side effect, the message delivery ratio is also increased.
Fast authentication methods in multiple operator maintained Wireless Mesh Networks

I have already introduced Wireless Mesh Networks in Section 1.2. I have found that authenticating mesh clients and controlling access to the network and its services are essential requirements for these networks. In addition, another important requirement in Multi-WMNs is the support of QoS-aware services and client mobility. This means that the authentication and access control mechanism should support the fast and seamless handover between network access points. The fact that the mesh routers are operated by multiple operators makes the issue more challenging.

The investigation of fast authentication methods in Multi-WMNs in general has been published in [Askoxylakis et al., 2009] and [Askoxylakis et al., 2010]. Two certificate based authentication methods have been proposed in [Buttyán and Dóra, 2009] and [Buttyán et al., 2010b] including the performance analysis and some additional proposals for constraint mesh clients.

In this chapter, I introduce a detailed list of requirements on the link layer authentication and access control enforcement in QoS aware multi-operator maintained mesh networks in Section 4.1. Then, in Section 4.2, I give an overview of the authentication and access control enforcement mechanisms proposed for Wi-Fi and mesh networks, and I analyze them with respect to the identified requirements. According to the result of the analysis, I propose two authentication and key exchange mechanisms and I explain their rationale in Section 4.3. In Section 4.4, I describe how to apply the two relevant standards in order to fulfill the requirements of Multi-WMNs. I propose and investigate in details a certificate based authentication mechanism in Section 4.5. In Section 4.6, I show the potential future research directions to improve my proposal and investigation. Finally, in Section 4.7, I summarize my work.

4.1 Requirements on authentication at the link layer

The main requirements for authentication and access control enforcement in a QoS aware multi-operator maintained mesh network can be classified into two groups: One concerning the authentication method and another one which is related to the establishment of the connection keys for the access control enforcement.

Requirements on the authentication method between the mesh client and the access point:
4. FAST AUTHENTICATION METHODS IN WIRELESS MESH NETWORKS

- **Fast authentication method to support user mobility:** As a main requirement, the authentication method has to support mobility of mesh clients who may use QoS aware services (e.g., VoIP). Such services may have requirements on the length of the interruptions in the communication that they can tolerate. When a mesh client moves from one access point to another, it has to re-authenticate itself as part of the handoff process. Before a successful authentication process, the mesh client should not be allowed to access the network (otherwise, it can exploit by changing the access points and gaining access without authentication). Thus, the re-authentication delay must be minimized in order to ensure that the interruption caused by the handoff remains tolerable for the applications.

- **Mutual authentication:** During the authentication method, the access point authenticates the mesh client, but the access point also has to prove its authenticity to the mesh client. If it is not the access point who authenticates the mesh client, then the mesh client also has to authenticate the third party (typically an authentication server).

- **DoS resistance:** The authentication method should not create any vulnerabilities to DoS attacks. Note, that a successful attack against a central unit (e.g., central authentication server) may lead to a state where no handoff can be completed.

- **Compatibility with standards:** In a multi-operator environment, it is fundamental that the protocols used in the authentication mechanism are standardized or built from standardized elements, otherwise a mesh client will not be able to authenticate itself to an access point belonging to another mesh operator.

- **Scalability:** One of the main advantages of mesh networks is the increased coverage. This, however, usually means a high number of mesh routers, access points, and mesh clients. Therefore, the authentication method must be scalable in terms of the number of access points and mesh clients.

- **No single trusted entity:** In a multi-operator environment, no single trusted entity may exist. Hence, each operator should run its own authentication server(s), but those could cooperate with the servers of other operators based on business agreements.

Requirements on the establishment of connection keys:

- **Connection keys should not reveal long term keys:** The connection keys that the access points obtain during the authentication of the mesh clients should not reveal any long-term authentication keys. This requirement must hold because in the multi-operator environment, the mesh clients may associate to access points operated by foreign operators.

- **Independence of connection keys:** As the neighboring access points may not trust fully each other due to the multi-operator environment, the authentication and the key generation mechanism have to prevent an access point from deriving connection keys that are used at another access point.

- **Freshness:** It must be ensured for both the access point and the mesh client that the connection key derived during the authentication process is fresh.

4.2 State-of-the-art and design options

4.2.1 Taxonomy

In the literature, many authentication and access control enforcement methods have been proposed. I categorize them by the place of the access control enforcement and by the place and type of the authentication.

The access control can be enforced at the following places:
4. FAST AUTHENTICATION METHODS IN WIRELESS MESH NETWORKS

- **Central access control enforcement**: The access control enforcement is done outside of the mesh network by a special entity in a centralized manner.

- **Access control enforcement at the border of the mesh network**: The access control is enforced by the gateways that are placed at the border of the mesh and the wired network.

- **Distributed access control enforcement**: The access control is enforced by the access points themselves.

If the access control is enforced by a central entity or at the gateways, then the system can not benefit from authenticating the mesh clients inside the mesh network. If the access control enforcement is distributed, the mesh client can be authenticated at the following network elements:

- **Remote authentication server**: In this case, the authentication servers of the operators are placed outside of the mesh network.

- **Local authentication servers**: In this case, the authentication servers are placed near to the access points within the mesh network, therefore, they can be reached by the access points within a few wireless hops.

- **Access points as distributed authentication servers**: In a totally distributed approach, the access points themselves function as authentication servers.

During the handoff, the authentication process can be initiated in a reactive or in a proactive manner:

- **Reactive authentication**: In this case, the authentication of the mesh client to the next access point and the establishment of the connection keys are carried out when the mesh client has already associated with the next access point.

- **Proactive authentication**: In this case, the connection keys are distributed to the potential next access point before the handoff process is started.

In addition, I classify proactive solutions by the participant who controls the distribution of connection keys:

- **Mesh client driven key distribution**: Before a mesh client performs a handoff, it creates security associations with the next or with each potential next access point.

- **Authentication server driven key distribution**: An authentication server distributes mesh client specific keys among the potential next access points such that the keys are available before the mesh client associates with the next access point.

### 4.2.2 Existing proposals

In Table 4.1, I categorized the proposed authentication methods found in the literature according to the above described taxonomy. In the following, I describe the categories in more details and I outline the main idea of the related proposals.

Note that the most of the proposed authentication procedures do not take into consideration the multi-operator environment. According to this, I consider the multi-operator environment only through the formerly defined requirements and I describe them in the single operator environment unless I state otherwise.

**Centralized enforcement of access control.** In an architecture where the access control enforcement is centralized, no authentication is required at the access points during the handoff process. The mesh client can associate to any access point, and the access control is enforced by redirecting the traffic of the mesh client to a central access control enforcement unit. The central unit makes forwarding decisions based on the origin of the traffic, typically, based on the MAC
and/or IP addresses of the mesh client. This solution is often used in Wi-Fi hotspots, for instance, using the Chilispot implementation [ChilliSpot, 2007]. The main drawback is that no connection key is established and an attacker can easily gain access by spoofing the MAC and IP addresses of an already authenticated device.

PANA (Protocol for carrying Authentication for Network Access) [Forsberg et al., 2008] is a general framework which can be adopted in centralized access control enforcement in the following way. The mesh client is authenticated only once, when it first associates with an access point. After a successful authentication, an IPsec tunnel can be established between the mesh client and a so called authentication agent, which relayed the authentication messages and obtained the connection key from the authentication server. As only the mesh client and the authentication agent can use this IPsec tunnel, this can be the basis of the access control enforcement.

The CAPWAP (Control And Provisioning of Wireless Access Points) standard [Calhoun et al., 2009b] supports centralized access control enforcement. The binding to the IEEE 802.11 standard is presented in [Calhoun et al., 2009a]. Herein, the physical and link level functionality of the access points are separated and the link level functionality is implemented in a central entity. This central entity communicates with a mesh client through a tunnel established between the central entity and the access point which the mesh client is associated with. During a handoff, the mesh client associates with the next access point and runs the 4-way handshake [IEEE Std 802.11i, 2004] with the central entity.

The main advantage of central access control enforcement is that no key material is stored in the access points. Hence, an attacker is not able to obtain any keys by compromising an access point. However, this architecture is extremely vulnerable to DoS attacks, because there is no possibility to deny the access before a message arrives to the central access control enforcement unit, and hence, an attacker can decrease the QoS level by injecting fake messages into the system. Another drawback is that the central unit is a bottleneck resulting in a potential scalability problem.

### Access control enforcement at the gateways

When the access control is enforced at the border of the wired and the mesh network, the mesh client can authenticate either to the gateway

<table>
<thead>
<tr>
<th>Central Access Control Enforcement</th>
<th>Border Access Control Enforcement</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Key distribution type</strong></td>
<td><strong>Proactive</strong></td>
</tr>
<tr>
<td><strong>Authenticator</strong></td>
<td><strong>Authentication server driven</strong></td>
</tr>
<tr>
<td><strong>Response</strong></td>
<td><strong>Mesh client driven</strong></td>
</tr>
<tr>
<td>Access points</td>
<td>[Chen et al., 2004; Aura and Roe, 2005]</td>
</tr>
<tr>
<td>Local AS</td>
<td>[Mishra et al., 2004]</td>
</tr>
<tr>
<td>Remote AS</td>
<td>[IEEE Std 802.11i², 2004; IEEE 802.11I-2008, 2008; Pack and Choi, 2002; Pack and Choi, 2004; Brik et al., 2005; Aboudagga et al., 2006]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Authentication Methods</th>
<th>[ChilliSpot, 2007; Forsberg et al., 2008; Calhoun et al., 2009a]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central</td>
<td>[Forsberg et al., 2008; Calhoun et al., 2009a]</td>
</tr>
<tr>
<td>Border</td>
<td></td>
</tr>
<tr>
<td>Distributed Access Control Enforcement</td>
<td></td>
</tr>
<tr>
<td>Key distribution type</td>
<td>Reactive</td>
</tr>
<tr>
<td>Access points</td>
<td>[Zhang and Fang, 2007; Chen et al., 2007]</td>
</tr>
<tr>
<td>Local AS</td>
<td>[Narayanan and Dondeti, 2008; Lopez et al., 2006]</td>
</tr>
<tr>
<td>Remote AS</td>
<td>[Calhoun et al., 2009a; IEEE Std 802.11i², 2004; Maccari et al., 2006a; Maccari et al., 2006b]</td>
</tr>
</tbody>
</table>

---

**Table 4.1: Categorized list of the proposed authentication methods**
4. FAST AUTHENTICATION METHODS IN WIRELESS MESH NETWORKS

or to a central authentication server. However, so far, no proposal exists where the gateway authenticates the mesh clients.

With mesh networks, the operators gain a cheap or feasible way of enlargement of the wireless radio coverage. However, the main objective remains to offer access to the Internet. From this point of view, the gateways can be good points to prevent the unauthorized access as it requires less administration.

The PANA protocol proposed in [Forsberg et al., 2008] can also be adopted in the case when the mesh client is authenticated to a central authentication server but the access control is enforced at the gateways. This is so because PANA allows for delegating the access control enforcement from an authentication agent to any other participant. Hence, a gateway can be an access control enforcement entity.

Although, the CAPWAP standard [Calhoun et al., 2009a] was not proposed for mesh networks, it may be adopted in the mesh environment to the case of access control enforcement at the gateway. In this case, the gateway may play the role of the central entity which runs the 4-way handshake with the mesh client.

These mechanisms improve the scalability of the centralized access control enforcement, but the DoS vulnerability described earlier still remains. Furthermore, if there are multiple gateways the problem of handoff between gateways must be solved.

Distributed access control enforcement with reactive authentication using a remote authentication server. A typical example of this case is the IEEE 802.1X [IEEE Std 802.1X-2001, 2001] authentication and access control model as described in the IEEE 802.11i standard [IEEE Std 802.11i™, 2004]. In this model, access control is enforced by the access points in a distributed manner. The client authenticates itself to a remote authentication server, which informs the access point about the result of the authentication, and also distributes a connection key (called PMK in the standard). This connection key (specifically the keys derived from it performing the so called 4-way handshake) is used to secure the follow-up communication at the link layer. A detailed description of the standard IEEE 802.11i can be found in Appendix A in Section A.1.

The messages of the authentication protocol are carried by the Extensible Authentication Protocol (EAP) [Aboba et al., 2004]. While many authentication protocols have been standardized in this framework (e.g., EAP-TLS, EAP-FAST, EAP-SIM), none of them are optimized for fast handoff. Recently, a new EAP method has been described for fast re-authentication in [Maccari et al., 2006a] and [Maccari et al., 2006b]. However, in this solution does, the connection keys are not independent.

I have already gave an overview of the CAPWAP standard [Calhoun et al., 2009a] when I described the distributed access control. Recall that the physical and link level functionality of the access points are separated in CAPWAP, and the link level functionality is implemented in a central entity. The CAPWAP standard has a special feature (not mentioned before in this section) that supports the delegation of the access control to the access points by sending the established connection key to them after a successful 4-way handshake performed between the mesh client and the central entity. Note that in this context, unlike in IEEE 802.11i, the connection key is not the PMK, because the access points only obtain keys derived from the PMK.

The main drawback of this approach is that the round trip time may increase significantly with the increasing distance (measured in wireless hops) between the access point and the authentication server. Hence, the round trip time can easily become higher than the round trip time that a QoS aware service can tolerate. Note that no application data can traverse the mesh network until the authentication is finished. Furthermore, the central authentication server is a single point of failure, which is vulnerable to DoS attacks.

Distributed access control enforcement with reactive authentication using local authentication servers. The problems listed above can be lighten by using local authentication servers placed close to the access points. Two EAP standard extensions in [Narayanan and Don-
4. Fast Authentication Methods in Wireless Mesh Networks

Deti, 2008; Lopez et al., 2006] are proposed to reduce the round trip time of the authentication messages by using local authentication servers placed between the access points and the central authentication server. The central authentication server is able to share the authentication key or a key derived from the authentication key with the local authentication servers. When an access point turns to any of the local authentication servers, that authentication server generates the connection key and sends it to the access point.

The main drawback of using local authentication servers is that those servers are within the mesh network where they may not be physically protected. Hence, it is hazardous to store long-term authentication information on them, as that information can be easily compromised.

Distributed access control enforcement with reactive authentication using the access points. The authentication is scalable and no preparations are required before the handoff when the authentication is performed between the mesh client and access points in a reactive way. However, other requirements may not be fulfilled as two proposals show.

The ID-based public-private key pairs can be used both for authentication and for key agreement with off-line central authority as it is exploited in [Zhang and Fang, 2007]. However, the private keys should be issued by the same central authority. Therefore, when a mesh client associates to a foreign access point it requires to have a temporary public-private key pair from the foreign operator for the key agreement or it can obtain one after an authentication process. In the latter case, fast handoff can not be guaranteed.

In [Chen et al., 2007], the authors suggest a change in the port-based network access control operation of IEEE 802.1X. Instead of restricting the mesh client to authentication messages through the uncontrolled port, the current access point allows mesh clients access to normal data traffic via a dynamically established tunnel between the current and the previous access point. The tunnel remains alive until the authentication is completed.

Distributed access control enforcement with server driven proactive authentication. In server driven proactive authentication methods, the authentication server is responsible for distributing connection keys prior to the handoff. Thus, when the handoff is taking place, the access points are able to make access control decisions locally without turning to the authentication server.

In [Mishra et al., Feb 2004], the connection keys are generated using the authentication key, the MAC addresses of the mesh client and the access point, and the connection key used at the current access point. The authentication server generates keys for the neighbors of the current access point and distributes among them. By neighbors, I mean the potential next access points that the mesh client may associate with. In this solution, the authentication server has to be aware of the location of the mesh clients, otherwise it is not able to determine which access points need keys next. A very similar idea is described in [Kassab et al., 2005] with some improvements: 1) the current AP sends the list of neighbors to the authentication server and 2) optionally, the current access point can distribute the current connection keys among the neighboring access points using IAPP protocol to postpone the connection key generation.

In [Bohák et al., 2007], the GSM authentication model was adopted to a Wi-Fi environment. The authentication server generates so called triplets which consist of some authentication information and a connection key. The triplets are sent proactively to the potential next access points that can use the authentication information therein to authenticate the mesh client performing the handoff, and the connection key for further access control enforcement. As the triplets are generated by the authentication server, the access points do not have to store long-term authentication keys. No concrete triplet distribution mechanism is proposed in that paper.

Distributed access control enforcement with mesh client driven proactive authentication. In contrast to server driven proactive authentication mechanisms, in the client driven case, the mesh clients themselves are responsible for getting the connection keys to the access points.
A mechanism called pre-authentication was proposed in the IEEE 802.11i standard [IEEE Std 802.11i™, 2004] that allows a mesh client to establish connection keys with the potential next access points prior to the handoff by performing full authentication through the current access point. The main advantage of this mechanism is that it is standardized and supports QoS aware services. However, the main drawback is that pre-authentication requires link level connection between the access points, and therefore, the mesh client can establish connection keys only with the one-hop neighbors of the current access point. Unfortunately, the set of potential next access points may not coincide the set of one-hop neighbors of the current access point.

In the IEEE 802.11r [IEEE 802.11r™-2008, 2008] standard, when a mesh client first connects to the network, it performs a full 802.1X authentication with a remote authentication server. The access point $\text{AP}_0$ through which this full authentication is performed will play a special role during the upcoming handoff processes. Before leaving the access point currently associated with, the mesh client indicates the handoff and the identity of $\text{AP}_0$ to the next access point (through the current access point or directly). The next access point obtains an authentication key $K$ from $\text{AP}_0$. The mesh client is able to generate $K$ using some public information and the initial authentication key shared with $\text{AP}_0$. The handoff is completed by running the 4-way handshake with the next access point and deriving connection keys from $K$. A detailed description of the IEEE 802.11r standard can be found in Appendix A in Section A.2.

The usage of multiple radio interfaces in mesh client devices was proposed in [Brik et al., 2005]. When multiple radio interfaces are available, one radio interface can be associated with a current access point and used for data traffic, while the other radio interface(s) can independently establish connection keys with other access points within radio range. The handoff then consists in swapping the roles of the radio interfaces: the radio interface which has already established a security association with the next access point becomes responsible for the data traffic, and the other radio interfaces(s) continues establishing security associations with new access points. Using multiple radio interfaces eliminates the problem that I identified in the case of pre-authentication, but this solution requires special hardware support (i.e., multiple radio interfaces) in the mesh client devices.

A solution is proposed in [Pack and Choi, 2002; Pack and Choi, 2004] for simplifying the connection key establishment between the mesh client and all the potential next access points. For this objective, the authors modified the key distribution mechanism of the IEEE 802.1X model. According to this modification, the mesh client and the authentication server establish a new connection key through the current access point, which is then distributed by the authentication server to the potential next access points. This approach is not compatible with the IEEE 802.11i standard, and it does not satisfy the requirement of independence of connection keys, because the new connection key is distributed among all the potential next access points.

Two ticket based approaches are introduced in [Aboudagga et al., 2006]. The idea is that after a full authentication, the authentication server generates tickets for each access point where the mesh client could move according to its mobility pattern. The tickets are delivered in one proposed solution to the potential next access points and in the other proposed solution directly to the mesh client. In the former case, the communication between the access points is based on the IEEE 802.11f protocol, also known as Inter Access Point Protocol (IAPP) [IEEE Std 802.11f™, 2003]. In the latter case, the mesh client sends the tickets to the access point at the time of the handoff. The tickets are encrypted using unique shared secrets between each access point and the authentication server. Therefore, the access points can obtain only those keys that are related to their own connections. The main drawback of this solution is the mobility prediction mechanism that has to be very precise, otherwise, no connection key may be established at the access point which the client wants to associate with. Furthermore, the IAPP protocol was withdrawn in 2006.

Distributed access control enforcement with proactive authentication to the access point. Instead of authenticating to a remote or local authentication server, in this category of solutions, the mesh client authenticates to the access point in a proactive manner.

There are three papers that follow this approach. In [Mishra et al., 2004], the authors propose
a solution where the currently used connection key is distributed to the potential next access points by the current access point — therefore it is an access point driven method —, and it is re-used there when the handoff takes place. The drawback is that this solution does not satisfy the requirement of independence of connection keys. In addition, the access points must trust each other even if they belong to different operators, which means that the requirement of no single trusted entity is not satisfied either.

In [Chen et al., 2004; Aura and Roe, 2005], the mesh client carries the new connection key in a credential that is sent to it by the current access point prior to the handoff. The credential is encrypted with a key shared between the current access point and the next access points. After associating with the next access point, the mesh client shows its credential, and the new access point decodes the connection key. Because of the time constraints, the authors propose to use symmetric keys cryptography to encrypt and decrypt the credentials. The authors also propose to run a full authentication after the lightweight credential based authorization. The mesh client can send data traffic parallel to the full authentication, hence, there are no constraints for the speed of the full authentication. The requirement of independence of connection keys is not fully satisfied in this solution either, because the previous access point generates the new connection keys. However, in this case, a full authentication is also carried out, therefore, this requirement remains unsatisfied only for a short period of time. The main drawback is that the mechanism as proposed does not fit any standards.

Generation of connection keys. Considering the generation of connection keys in the various proposals, the connection keys are computed using the following data (or some part of them): the authentication key, the previous connection key, public information of the access point, some random numbers. Table 4.2 shows what requirements are fulfilled by the different input data. Note that during the computation of the connection keys, these input data can be combined. However, the combination must ensure that the access points are not able to obtain the authentication key from the computed connection keys. Besides the appropriate key generation process, the independence of the connection keys can be fulfilled by performing a full authentication after the completed fast handoff.

<table>
<thead>
<tr>
<th></th>
<th>Ensure freshness for the mesh client</th>
<th>Ensure freshness for the access point</th>
<th>Independence of connection keys</th>
<th>Long term key protection</th>
<th>Mutual authentication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication key</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Previous connection key</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Public information of AP</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Random number from AS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Random number from MC</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

* AP – Access point
† AS – Authentication server
‡ MC – Mesh client

4.2.3 Summary

In Table 4.3, I summarize how the various approaches for authentication and access control enforcement described above satisfy the requirements identified earlier. Unfortunately, it is unambiguous what compatibility of a whole category with standards means. I indicate that a category is compatible with standards if at least one method found in literature is a standard or based on a standard
and the standard is not in draft version. Note that the status of the compatibility can quickly change with new accepted standards or new proposed methods.

Table 4.3: Requirements and authentication methods

<table>
<thead>
<tr>
<th>Authentication method</th>
<th>Fast (re)authentication method</th>
<th>DoS resistance</th>
<th>Compatibility with standards</th>
<th>Scalability</th>
<th>No single trusted entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central access control enforcement</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Boundary access control enforcement</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Reactive Remote auth. server</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Reactive Local auth. server</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>AS driven† proactive Authentication server‡</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mesh client driven, proactive Authentication server‡</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Reactive Access point</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>AS driven† proactive Access point</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Mesh client driven, proactive Access point</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
</tbody>
</table>

* ACE – Access control enforcement
† AS – Authentication server
‡ Remote or local authentication server

When access control is enforced at a central entity or at the border of the mesh network, the system is not able to deny the forwarding of packets inside the mesh network coming from unauthorized mesh clients. Therefore, these methods create DoS vulnerability in the network. Furthermore, in the case of central access control enforcement, the network is not scalable, because the central access control enforcement unit becomes a bottleneck.

When a central authentication server is used with reactive authentication, the round trip time of the message exchanges of the authentication protocol can be too long such that the QoS aware services cannot tolerate that. Besides that, if the authentication server is DoS attacked, no authentication can be performed during the handoff in the entire network. These problems are solved when local authentication servers are used, but then the problem is that those servers reside in the mesh network and they can be attacked and compromised physically.

Distributed access control enforcement with proactive authentication methods satisfy all the requirements. However, not all parts of the connection key distribution process is handled in a standardized way when the key distribution process is server driven. In the case of mesh client driven proactive authentication, the proposed mechanisms often require conditions that are difficult to satisfy (e.g., multiple radio interfaces in mesh clients).

The requirement of no single trusted entity is not satisfied when the previous access point authenticates the mesh client during or before the handoff, because an access point must trust the previous access point as an authenticator even if it belongs to another operator.

### 4.3 Selected approaches and their rationale

After getting an overview of the current solutions, I decided to propose two different methods. One method relies on two standards, in particular, I propose a method to use two standards jointly in
order to satisfy all the requirements described above. I could not evaluate this method because of the lack of the implementation of a standard. I also proposed a certificate based method which assures local authentication at access points. The latter method is investigated exhaustively.

The two most relevant standards are the IEEE 802.11r and the Handover Keying (HOKEY) IETF standard. The main design principle was to support fast handover in the IEEE 802.11 environments and in EAP protocol based authentications, respectively. These protocols can be considered as the fast handover methods of the future because they are supported natively in Wi-Fi devices. However, the standard IEEE 802.11r does not support the inter-domain handover, while the HOKEY desires to turn to an authentication server during the handover which may cause large delays. But what is missing in one standard is very well solved in the other one. These protocols were designed in such a way that they can complement each other fitting the multi-operator maintained QoS-aware mesh networks. I give an overview of both standards (and of the IEEE 802.11i which the IEEE 802.11r relies on) in Appendix A. In the next section, I show how to design the authentication mechanism for a Multi-WMN using the above mentioned two standard.

Beyond the standard solutions, I propose a certificate based authentication and key exchange method which has some benefits compared to the symmetric key based solutions described in the above mentioned standards. Public key cryptography fits well the multi-operator environment because the mesh clients can be authenticated to access points locally even if the access points belong to other operators. The authentication can be performed between the mesh clients and the access points in a distributed manner. The main benefit of the distributed method is that the authentication architecture is very scalable, eliminates the single point of failure and mitigates the effect of some DoS attacks, unlike a centralized approach where the authentication server can be a potential target of an attacker preventing any handover in the network. The main drawback of a certificate based authentication scheme is the time consumption of the public key cryptographic primitives. However, I propose protocols that carefully take into account the complexity of the public and the private key operations, and in this way, they reduce time consumption such that a QoS aware service can tolerate the delay even if the access points are limited in computational power.

### 4.4 IEEE 802.11r and HOKEY in a multi-operator environment

I gave a short overview of the IEEE 802.11r and IETF HOKEY standards in Section A.2 and A.3, respectively. Both standards support the fast handover using fast authentication methods, however, none of them fulfill entirely the requirements defined for multi-operator environments. The IEEE 802.11r does not support the inter-domain handover and in the HOKEY standard the mesh client has to communicate with authentication server during the handover which may cause delay. The two standards do not substitute but complement each other as the RFC 5169 [Clancy et al., 2008] states, too, however it is not detailed how to integrate the two mechanisms. In this subsection, my objective is to show in a deeper level how the two mechanisms can complement each other in a multi-operator maintained QoS-aware mesh network.

#### 4.4.1 Architecture

In Figure 4.1, I show a scenario through I explain how I combine the two standards in Multi-WMNs. The access points which are placed near to each other and belong to the same operator form a domain. In each domain, each operator appoints an access point to play the role of the local authentication server (LAS). The domains and LASs must be determined in such a way that any access point should reach the nearest LAS belonging to the same operator in few wireless hops.
4.4.2 Initial authentication

When a mesh client (subscriber at operator $O_1$) associates first to the mesh network, it performs an initial authentication. The initial authentication is not time critical operation, therefore a time consuming authentication with a remote authentication server (RAS$_1$) can be performed. An arrow marked with $\star$ shows the path of the initial authentication in Figure 4.1 in the case when a mesh client associates first to a foreign access point (AP$_{21}$). AP$_{21}$ forwards the authentication messages to the dedicated LAS$_2$. LAS$_2$ recognizes that the message belongs to a full authentication method and forwards to the dedicated remote authentication server (RAS$_2$). Finally, RAS$_2$ forwards all the authentication messages to the home authentication server (RAS$_1$).

LAS$_2$ receives ‘EAP Response/Identity message’ if the mesh client initiates full authentication and ‘EAP Initiate Re-auth Start’ message if the mesh client performs an inter-domain handover. RAS$_2$ obtains only initial authentication messages. The first EAP message contains the identity of the mesh client which consists of the unique name at the home operator and the operator’s name (usually called domain name, but in my case a group of access points form a domain and maybe more domains belong to an operator). RAS$_2$ forwards the authentication messages according to the operator’s name. In the last authentication message RAS$_1$ sends the MSK to AP$_{21}$.

After AP$_{21}$ gets the MSK, it starts to play the role of R0KH and calculates PMK-R0 and PMK-R1. The mesh client plays the role of S0KH and calculates the same keys. Finally, AP$_{21}$ and the mesh client plays the role of R1KH and S1KH, respectively and they perform the FT 4-way handshake.
4. FAST AUTHENTICATION METHODS IN WIRELESS MESH NETWORKS

4.4.3 Inter-domain handover
After a successful authentication, RAS\(_1\) delivers domain specific keys to the foreign RASs (RAS\(_2\) and RAS\(_3\)) and all RASs forward to the appropriate LASs (LAS\(_1\), LAS\(_2\) and LAS\(_3\)) as arrows marked with \(\oplus\) show in Figure 4.1. The domain specific key is the DSRK defined in the HOKEY standard. The DSRK is utilized when the mesh client performs inter-domain handover. Note that these keys are derived from the key generated during the initial authentication using one-way function. Therefore, the domain specific keys are short-term keys and an attacker is not able to reveal any long-term keys if a LAS become compromised.

Note that no DSRK is needed to deliver into the domain where the mesh client first associated to because there, the MSK is delivered and no further keys from the remote authentication server are required for the intra-domain handover. The delivery of MSK does not cause any security flaws, because the DSRK is derived from EMSK which is generated cryptographically separated from MSK.

The DSRK is calculated as it is shown in Eq. (A.6). Herein, the domain name is the MDID (Mobility Domain Identifier) defined in the standard IEEE 802.11r. The MDID is propagated by the access points, therefore, the mesh client is able is calculate any DSRKs.

When the mesh client associates to a new mobility domain, the first access point AP\(_{11}\) may have no information about the mesh client. In that case from the access point of view, it needs to perform a full authentication with the mesh client. However, this authentication is handled by LAS\(_1\), as the arrow marked with \(\oplus\) shows. LAS\(_1\) obtained the DSRK when the mesh client associated to the mesh network.

The mesh client calculates the DSRK obtaining the MDID and derives rRK (see Eq. (A.7)) and rIK (see Eq. (A.8)) as LAS\(_1\) does. Using rIK, the mesh client performs a HOKEY re-authentication with LAS\(_1\) as it is shown in Figure A.8. Herein, the authenticator is AP\(_{11}\) and the Foreign AS is the LAS\(_1\). The value of the SEQ is 0 and rMSK is calculated over this value as no other key derivation is desired for the intra-domain handover.

At the end of the local authentication, AP\(_{11}\) obtains rMSK from LAS\(_1\). Both parties complete the authentication by playing the roles defined in standard IEEE 802.11r.

4.4.4 Intra-domain handover
The intra-domain handover works according to the standard IEEE 802.11r. The access point AP\(_{11}\) where the mesh client first associates in a mobility domain becomes the R0KH of the mesh client. The R0KH as the standard IEEE 802.11r states is responsible for deriving keys for other access points in the mobility domain.

Before the mesh client associates to the next access point AP\(_{12}\), the mesh client notifies it in advance to AP\(_{12}\) according to the protocol introduced in Figure A.5. AP\(_{12}\) has time to obtain the relevant PMK-R0 key from AP\(_{11}\).

I did not show here, but in a scenario other than presented in Figure 4.1, it can happen that a LAS\(_i\) become R0KH, too. In that case, the inter-domain handover will be very quick as no communication with other entities is required.

4.4.5 Fulfillment of the requirements
Here, I follow the list of requirements defined in the beginning of this chapter and I show that my architecture designed according to IEEE 802.11r and HOKEY standards fulfills these requirements.

- **Fast authentication method to support user mobility**: Both the IEEE 802.11r and HOKEY standard was designed for supporting seamless handover.

- **Mutual authentication**: The re-authentications are based on shared secrets derived from the result of the initial EAP authentication. If this EAP authentication provides mutual authentication, the re-authentication mechanism does, too.
DoS resistance: There are three different authentication server entities in my proposed architecture. Even if an attacker dismisses the remote authenticator, all the currently associated mesh clients can perform handovers. If an attacker dismisses an R0KH access point, the mesh clients can choose another R0KH by re-authenticating to local authentication server. If a local authentication server is attacked, no new mesh clients can associate to the access points belonging to the same domain. As one can see, the mechanism is robust because of the redundant architecture.

Compatibility with standards: The IEEE 802.11r and HOKEY are standards and my architecture was designed to correspond to these standard.

Scalability: There are three different authentication server entities in my proposed architecture: 1) the central authentication server which is responsible for initial authentications, 2) the local authentication servers which are responsible for the first authentication in their mobility domain, and 3) the R0KHS which are responsible for fast re-authentications in the mobility domains. Note that there is no dedicated R0KH in the mobility domain. The R0KHS may vary on the mesh clients because that access point becomes the R0KH of a specific mesh client where that mesh client first associated to the mobility domain. The hierarchical and distributed authentication scheme provides scalability.

No single trusted entity: No single trusted entity is required, each operator maintains its own infrastructure elements. And only central authentication servers need to communicate with each other according to the bilateral service agreements.

Connection keys should not reveal long term keys: Each key is derived from another with one-way function, therefore it is not feasible to reveal any long term keys.

Independence of connection keys: Both the DSRKs and PMK-R1 keys are generated cryptographically separated and each key is delivered only to the authorized entity.

Freshness: The lifetime of the connection keys is no longer than the lifetime of MSK and EMSK. The lifetime of the MSK and EMSK are also very short as it is only a temporary key generated during the full authentication.

4.4.6 Implementation issues

As I showed, the standards IEEE 802.11r and HOKEY can be combined to have a fast handover scheme supporting the multi-operator environment. An implementation of the proposed architecture requires the execution of both IEEE 802.11r and HOKEY. The IEEE 802.11r standard, referring to the mesh client and the access point, is already implemented in wpa supplicant and hostapd [Malinen, 2009], respectively. HOKEY has to be implemented in each participant: in the mesh client, in the access point and in the authentication server(s). So far, I have not found any implementations of the HOKEY protocol.

4.5 Certificate based authentication and access control

As I have stated in Section 4.3, using public key cryptography is a suitable approach in multi-operator maintained mesh networks. However, the critical point of using asymmetric based crypto in the authentication and key establishment mechanism is the computational delay. In this section, I propose two certificate based authentication protocols for Multi-WMNs. First, I describe the architecture of the certificate based authentication protocols. Then, I investigate the speed characteristic of some classical cryptographic primitives. After introducing a nonce-based and a timestamp-based authentication method, I define what public key algorithms and key sizes to use during the authentication in order to fulfill the general security requirements while still ensuring a short authentication delay during the handover. I also investigate the authentication delay in real environment.
4. FAST AUTHENTICATION METHODS IN WIRELESS MESH NETWORKS

4.5.1 Architecture

In my certificate based authentication and access control scheme, each operator operates its own certificate authority (CA). Each CA is responsible for issuing certificates for the access points belonging to the operator and issuing certificates to their subscribers. The CA also maintains the certificate revocation list (CRL).

The operators which decide to cooperate (O_1 and O_2) issue cross-certificates of their CAs which means that operator O_1 issues a certificate on the public key of O_2’s CA and O_2 issues a certificate on public key of O_1’s CA. With the cross-certificates, entities (subscribers or access points) can perform certificate based authentication and key exchange mechanisms even if they belong to different operators.

Each certificate must contain the following items:

- Identity of the issuer
- Time of issuance
- Lifetime (or time of expiration)
- Identity of the owner
- Key usage (encryption or digital signature)
- Public key algorithm
- Owner’s public key
- Certificate signature value

The X.509 format [Cooper et al., 2008], as it is a standard format, makes the communication between foreign entities smoother and it is prepared to be extended with additional items. The disadvantage of this format is that it may waste a lot of space.

The certificate signature algorithm consists of two parts: 1) definition of the hash algorithm and 2) definition of the digital signature algorithm. No special requirements are on the hash algorithm beyond the fundamental security related ones (e.g., collision resistance) because it is usually a very fast cryptographic primitive. In contrast to this, the digital signature is a more time consuming one. Herein, the RSA algorithm is a perfect solution, because even if the signing operation needs considerable time, it is not performed in a time critical period. On the other hand, the verification, which is performed during the time critical handover, is very fast.

I suggest to handle the revocation in different ways depending on whether a certificate is issued to a mesh client or an access point. Maintaining CRL suits very well access points because they have permanent connection to the CA. In contrast to this, the mesh clients, who can be off-line while the private key of an access point becomes compromised, are not able to download the CRL before it connects to the mesh network. Therefore, the CA maintains the CRL and distributes it among the access points and CAs belonging to other operators regularly or when the list changes. The CRL contains the revoked public key pairs of the mesh clients whose certificates are issued for longer time period (months or years). In contrast, the access points’ certificates are short-term, valid only for some days. The access points are able to renew their keys and certificates at any time, because they are part of the infrastructure and they are always on-line. The size of the vulnerability window is small due to the limited lifetime, and the damage is also smaller in case of a compromise.

4.5.2 Design rationale

Here, I investigate the properties of the public key based cryptographic algorithms based on published benchmarks [Page et al., 2008] and own measurements (see Section B). I considered the following key exchange, digital signature and encryption algorithms: Diffie-Hellman (DH), Elliptic Curve DH (ECDH), RSA, DSA, EC-DSA, EC-ElGamal.
4. FAST AUTHENTICATION METHODS IN WIRELESS MESH NETWORKS

Benchmarks showed that the elliptic curve based solutions (ECDH, EC-DSA, and EC-ElGamal) are not beneficial because these algorithms are slower than the classical ones at similar security levels. In the case of DH key exchange algorithm, the computational complexity is as large as the private key operation of RSA, but on both sides. Furthermore, DH does not provide authenticity, and the key exchange and providing authenticity all together would cause too long delay. Therefore, in what follows, I consider only the RSA and the DSA algorithms.

In the case of RSA, the public key operations (encryption and digital signature verification) are quick operations when the exponent is relatively small (typically 65537), while the private key operations (decryption and digital signature generation) are three orders of magnitude slower. In contrast to this, the digital signature generation with DSA with some precalculation can be performed very quickly, while the verification is three orders of magnitude slower. In what follows, I assume that the DSA precalculations are performed and the generation of digital signature is fast.

The latency of a public-key cryptographic operation on one block mainly depends on the key size of the algorithm and on the performance of the device which performs the algorithm. There is always a trade-off between the speed of the algorithms and the level of the security. Nowadays, e.g. RSA with 512 bit key size is secure for 1 hour, and with 1024 bit for 1 year [Nåslund, 2008]. In my proposals, I consider only these two key-sizes because the operations with 256 bit long or shorter keys are insecure and with 2048 bit long or longer keys cause intolerable delays in the authentication process.

4.5.3 Certification based authentication and key transport protocols

Nonce based solution

In [Buttyán and Dóra, 2009], I chose the Blake-Wilson and Menezes Provably Secure Key Transport Protocol [Blake-Wilson and Menezes, 1998] (BWM), because of two reasons. Firstly, among the considered protocols [Boyd and Mathuria, 2003] this protocol has the minimal number of public key based computations as one signature per each participants, that the protocol requires, is a minimum to prove that each one is online and a public key based cryptographic primitive to provide a secure key for the upcoming communication. Secondly, this protocol was proven to be secure [Blake-Wilson and Menezes, 1998].

However, the BWM protocol as I could adapt it to the mesh environment has a DoS vulnerability. Namely, the AP has to prove its presence first which requires public key cryptographical computation on MC side and therefore, a malicious MC can perform a DoS attack against the AP, easily. Since the key has to be transported by the AP (the motivation is explained in this section later on), the roles cannot be changed easily. Therefore, I changed only the order of the verification of online presence.

The procedure of my nonce based authentication mechanism is shown in Figure 4.2. AP first sends its ID, and a fresh nonce \(N_{AP}\). MC also generates a nonce and concatenates it to the ID’s of the participants \(ID_{AP}\) and \(ID_{MC}\) and the nonce generated by AP. The signature \(S_{P_{MC}}(M_1)\) is calculated on these data using MC’s private key. One certificate issued by the operator of the MC \(OP_{MC}\) for the digital signature \(Cert_{OP_{MC}}(S_{MC})\) and one for the encryption \(Cert_{OP_{MC}}(Q_{MC})\) is included in the message, too. On the other side, AP verifies the signature and the certificates and checks whether \(N_{AP}\) and \(ID_{AP}\) is the nonce and the identity, respectively that it sent in the first message. Then, AP encrypts the previously generated key \(K_{AP}\) concatenated with its ID using \(Q_{MC}\). AP concatenates the IDs, the nonces and the encrypted key and calculates its digital signature \(S_{P_{AP}}(M_2)\). The third message consists of the concatenated data, the digital signature and the certificate issued by the AP’s operator \(OP_{AP}\) for generating digital signatures. Finally, MC has to verify the IDs and nonces match with the ones previously sent and received, and it verifies the certificate, too. After decrypting \(K_{AP}\), MC has to check whether the ID sent in encrypted text matches \(ID_{AP}\) and also matches the identity sent in the certificate.

The connection key \(K_{conn}\) is calculated with the following method:

\[
K_{conn} = \text{Hash}(K_{AP}, N_{MC})
\]
4. FAST AUTHENTICATION METHODS IN WIRELESS MESH NETWORKS

Figure 4.2: Nonce based authentication

where \( Hash() \) is a one-way function.

The protocol assures for both participants that they are online. Each participant generates a
digital signature over a nonce sent by the other party. The nonce has to be fresh and unpredictable.

Implicit key authenticity is assured, because the key \( K_{AP} \) is known only by the AP, who
calculated the random bits, and MC, the only who can decode the message sent by AP and
encrypted with the public part of \( Q_{MC} \). The \( Q_{MC} \) is used only for encrypting \( K_{APs} \), and only the
MC is able to decrypt with the private key. As no else than the AP and the MC knows the \( K_{AP} \),
only they can calculate the \( K_{conn} \).

Key freshness is assured, because \( K_{conn} \) is calculated from two elements provided by both
participants using one-way function. One-way function assures that AP is not able to choose a
\( K_{AP} \) and MC is not able to choose a \( N_{MC} \) such that \( K_{conn} \) takes a desired value.

The protocol itself does not provide key confirmation, but my implementation will rely on
standard IEEE 802.11i [IEEE Std 802.11i\(^\text{TM}, 2004\)] which provides key confirmation through the
4-way handshake.

**Timestamp based solution**

I propose a timestamp based solution, too. Note that in this case no new requirement has to
be met because the verification of the certificates requires loosely synchronized clocks, anyway.
Furthermore, it needs fewer random bits and the signed timestamps can be used as a basis of
accounting (however this was not mentioned as a requirement before).

The timestamp based scheme, which uses two digital signatures and one encryption, can be
seen in Figure 4.3.

First, MC sends its timestamp \( t_{MC} \) signed with its private key. The first message contains
the IDs of the participants and the relevant certificates \( (Cert_{sMC}) \). After AP has checked if the
difference between \( t_{MC} \) and \( t_{AP} \) (i.e. AP’s currently generated timestamp) is within the acceptance
time window and the IDs are correct, it verifies the signature and the certificates. The acceptance
time window is the maximum difference between the timestamps sent in messages \( (t_{AP} \text{ and } t_{MC}) \)
and current time that a participant can accept. AP creates a message containing the IDs, \( t_{AP} \) and
an encryption of a securely generated key \( K \) using MC’s public key. The message sent back to MC
contains a signature over these data and the hash value of the message received from MC. The
4. FAST AUTHENTICATION METHODS IN WIRELESS MESH NETWORKS

**Figure 4.3: Timestamp based authentication**

relevant certificates ($Certs_{AP}$) are also included. MC verifies the signature and the certificates, and checks the difference between the clocks. If the IDs agree with the value sent in the first message, MC decrypts key $K$.

AP’s DoS resistance is provided by the fact that the MC sends the first authenticated message and the AP has to generate and encrypt $K_{AP}$ and generate digital signature only after MC proves its authenticity and the run of the protocol will end successfully with high probability. Although an attacker can replay eavesdropped messages, it is limited to those messages which are sent to the current AP within the acceptance time window.

This scheme, as it has been presented so far, provides key authenticity and key freshness both for MC and AP, but no key confirmation. The key is controlled by both parties as it is calculated in the following way:

$$K_{conn} = Hash(K_{AP}, t_{MC})$$  (4.2)

The protocol assures for both participants that they are online. Each participant generates a digital signature over the current time. An attacker is not able to get a valid digital signature over a future timestamp, only if the time is not synchronized correctly.

The key confirmation is provided by the 4-way handshake of standard IEEE 802.11i [IEEE Std 802.11i™, 2004]. Implicit key authenticity is assured, because of the same reasons described at description of the nonce based solution.

Key freshness is also assured, because $K_{conn}$ is calculated from two elements provided by both participants using one-way function. One-way function assures that AP is not able to choose $K_{AP}$ and MC is not able to choose $t_{MC}$ such that $K_{conn}$ takes a desired value. Note that $t_{MC}$ is predictable, in contrast to the nonces, but the calculation of $K_{AP}$ to get a specific $K_{conn}$ is not feasible if the entropy of $K_{AP}$ and $K_{conn}$ are high enough. I suggest to use 128 bit long keys and SHA-1 hash function as a one-way function.

### 4.5.4 Public key algorithms and key parameters

So far, I did not investigate the parameters of the public key algorithms and the certificates. In both protocols, a MC needs a public key pair for the encryption ($Q_{MC}$) and another one for the digital signature ($P_{MC}$). APs only require a public key pair for digital signature ($P_{AP}$).
Note that a MC has two different public key pairs: 1) one for encryption and 2) one for digital signature. It is insecure to use the same key pair for the two function, because an attacker can exploit one function against the other. Note that it requires two different certificates when the certificates are issued according to X.509 standard as well as I suggested because of compatibility reasons.

In order to decrease the latency of the verification of two certificates owned by an entity, I define an extension for the X.509 certificates as the standard is flexible enough to add new entries. Considering certificates $A$ and $B$, when a CA issues the certificates, it generates certificate $B$ in the regular way, but it calculates its hash value $h$, too. $h$ is added to certificate $A$ as an X.509v3 extension and when the digital signature is calculated it includes the $h$, too. If a verifier can handle this extension, the verifier calculates the hash value of certificate $B$ and compares it with the appropriate extension of certificate $A$. If it matches, the verifier accepts certificate $A$, if not, the certificates are rejected. With this mechanism, the two certificate verification can be reduced to one signature verification and one hash value computation. If a verifier does not support this extension, the verifier can simply ignore it and verify the two certificates separately.

Regarding the digital signatures and encryption, it is beneficial to shift as many computationally intensive operation to the MC as many possible, because of the following reasons:

- Usually MCs that benefit from the seamless handover are more powerful than the APs. It is because one of an important design principle in the case of MCs are to handle media streams which are, therefore, usually equipped with powerful elements. On the other hand, an important design parameter is the price in the case of APs, therefore, APs are usually constrained devices.

- When the authentication of MCs at an AP are overlapping, the longer lasts the authentication at the AP side, the longer the other MCs have to wait. Furthermore, the more the AP has to calculate, the bigger the chance is that more authentications are overlapping.

- Finally, if the MC has to compute more, it increases the DoS resistance, as an attacker needs more investment to perform a successful DoS attack.

Considering the encryption, currently RSA is a widely known and accepted algorithm which is asymmetric from the time consumption point of view. As I already described, the public key operation of RSA (encryption) is quicker and the private key operation (decryption) is slower operation. This is the reason that I suggest the AP to generate and encrypt the secret key $K_{AP}$ used for connection key.

In order to ensure the confidentiality of the key $K_{AP}$, I propose to use minimum 1024 bit long keys [FP7 ECRYPT II, 2007].

Regarding the parameters of AP’s and MC’s public key used for digital signature, I differentiate two cases: 1) when the MC is significantly more powerful than the AP and 2) when the difference is less significant. I describe these two cases in the following subsections.

**Powerful mesh client**

When the MC has more power than the AP (which is the typical case if I consider laptop computers as MCs), the MC can use RSA for digital signature, while the AP generates digital signatures with DSA. In that case all the computationally intensive operations (private key operations with RSA and digital signature verification with DSA) are shifted to the powerful MC, whereas, the lightweight operations are performed by the AP.

The public keys of the MCs, as I defined earlier, are long-term keys. Therefore, I chose 1024 bit long public-private keys. The APs’ public key are mid-term as they may change them frequently (e.g. daily). I also chose 1024 bit long keys for mid-term keys.
4. FAST AUTHENTICATION METHODS IN WIRELESS MESH NETWORKS

Constrained mesh client

Note that a less powerful MC is not able to perform all the computing intensive operations. Therefore, I propose another technique to reduce the delay of the whole protocol at the cost of some pre-computation by both participants.

The idea is based on speeding up the digital signature operations using weak keys. The certificates belonging to weak keys have a very short lifetime, such that they surely expire by the time they will be broken.

The weak keys and the belonging certificates are generated by the participants before the handover happens. In fact, MCs and APs issue certificates themselves. I have to emphasize that these certificates are not self-signed certificates but new elements of certificate chains generated by a MC or an AP. Let us assume that MC wants to issue a short-term certificate. First, it generates a weak public key pair \((T_{MC})\). Then, it uses its identity as the name of the certificate and determines the expiration date which must be defined carefully, as the weak key can broken quickly. Finally, it supplies the certificate with digital signature using its private key \(C_{MC}\), which is certified by the MC’s operator for issuing certificates for weak keys. Therefore, any other entity who knows the CA’s public key can validate the authenticity of the weak public key. The same mechanism can be performed at the AP side.

The validity of the certificates are short-term, therefore, maintaining of CRL is not required for implementing this mechanism. Furthermore, in this mechanism, the target AP and the MC which will perform the handover do not need to communicate with each other or to obtain some information about each other, because the certificates are issuer specific. The certificates of the weak keys are signed with RSA so they can be verified very quickly.

I suggest to use 512 bit long keys as short-term keys which seems to be the best tradeoff for my purpose today between the validity time and the computational overhead. Similarly to the case of a powerful mesh client, the MC uses RSA and AP uses DSA to generate digital signatures.

The time synchronization needs to be performed in a secure way, otherwise an attacker can make a MC or AP to accept an already expired certificate of an already broken public key pair. However, the investigation of the secure time synchronization is out of scope of this thesis.

Even if a secure time synchronization is provided by the system, it cannot be performed before the first association to the network. Note that in that case no QoS aware services run by the MC, therefore, any authentication method is suitable which does not require synchronized clocks.

In Figures 4.4 and 4.5, respectively the nonce based and timestamp based authentication scheme is described using weak keys. Here, I emphasize the differences compared to the basic protocols shown in Figures 4.2 and 4.3.

As Figure 4.4 shows, MC and AP generate weak keys and certificates. MC must check before the handover whether it has a valid certificate. If not, it generates a new one. AP must have a valid temporary key at any time, because the AP does not know when the next MC wants to authenticate. Therefore, it always generates a new certificate before the previous one expires.

The implementation should be designed such that the MC and the AP always have a weak key and belonging certificates available during the handover process. Nevertheless, the participants can use their public-private keys that is used for issuing certificates for weak keys or dedicated public-private keys and belonging certificates should be maintained to handle this case. Obviously, the fast handover can not be assured in this case.

In the authentication phase, the digital signatures are generated using the temporary private keys. Instead of the certificate of the long-term public key used for digital signature (e.g. \(Cert_{OP_{MC}}(P_{MC})\)), each participant includes two certificates: 1) The short-term certificate for the temporary key used for digital signature (e.g. \(Cert_{MC}(T_{MC})\)), and 2) the certificate of the long-term key which is used for issuing short-term certificates (e.g. \(Cert_{OP_{MC}}(C_{MC})\)).

At both sides, the participants have to verify the whole certificate chain, which requires one more certificate verification compared to the case when no weak key is used.

Note that the usage of the weak key mechanism is optional for each MC. The AP uses the weak key mechanism only if the MC used weak key mechanism. Otherwise the powerful case is supported. It is important because, as one can learn from the performance analysis, the weak key
mechanism may increase the authentication delay when the MC is powerful. Consequently, the APs must have public-private keys and belonging certificates for both cases.

In the paper [Buttyán et al., 2010b] that this section relies on, the timestamp based protocol is modeled in a timed process algebra suitable for timing issues (tCryptoSPA). A possible methodology is shown to carry out an analysis with respect to the use of the weak keys. In particular, the analysis with respect to the use of the weak secret key of MC is present. The method can be opportune used to analyze also the correct use of the weak key of AP. It is proven that if MC uses weak keys generated by itself with short-term certificates for digital signature, it is as secure as the timestamp based protocol with long-term keys where the long-term keys can be revealed with a very low probability.

Note that the usage of the weak key mechanism is not limited to the considered authentication scenario. It can be beneficial where the usage of the public key cryptography is advantageous, but the devices are constrained. If weak keys are used among stationary devices, the performance improvement can be more significant, because the certificate verification is performed only once in its lifetime.

### 4.5.5 Cross-certificates

As I have already mentioned, my solution is designed for multi-operator environment. In such an environment, the operator of the AP and the MC may not be the same. Therefore, their root CA is different. To handle these situations, the root CAs issue so called cross certificates. In that case, the cross certificates are sent with the other certificates. These enlarge the size of the sent
VERIFIES THE TIMESTAMP, THE SIGNATURE AND THE CERTIFICATES

\[ \text{REQ} = \text{[M = [ID}_{MC}, \text{ID}_{AP}, t_{MC}], S_{MC}(M), \text{Cert}_{MC}(T_{MC}), \text{Cert}_{OP_{MC}}(C_{MC}), \text{Cert}_{OP_{MC}}(Q_{MC})} \]

\[ \text{RESP} = \text{[ID}_{AP}, \text{ID}_{MC}, t_{AP}, E_{OP_{MC}}(K_{AP}), S_{AP}(\text{Hash(REQ)|RESP)}, \text{Cert}_{AP}(T_{AP}), \text{Cert}_{OP_{AP}}(C_{AP})} \]

1. Verifies the timestamp, the signature and the certificates
2. Decrypts \( K_{AP} \)

**Figure 4.5:** Timestamp based authentication with weak key mechanism

messages and also requires one further certificate verification. The cross certificates are not shown in Figure 4.2, 4.3, 4.4, and 4.5.

Note that in a regular case, the cross certificate does not change frequently and the participants can learn it, in particular the public key of the other operator’s CA can be learnt. Thus, the authentication delay after some bootstrap time become the same as if the MC would be authenticated at an AP at the same operator. However, in the following, I will investigate a cross certificate scenario as a worst case scenario, too.

The verification of the certificate chains must be limited to the operators’ CA that has directly signed the certificate of the operator which issued the certificate of the MC. Otherwise, the MC could connect to APs of an operator that has no agreement with the operator of the MC, but both operators have an agreement with a third operator.

### 4.5.6 Performance analysis

**Implementation**

I created a proof-of-concept implementation. I embedded the authentication messages into EAP (Extensible Authentication Protocol) frames [Aboba et al., 2004]. EAP messages are embedded into EAPOL messages in IEEE 802.1X [IEEE Std 802.1X-2001, 2001] which is referred by IEEE 802.11i and IEEE 802.11r, the current standard solutions for Wi-Fi authentication. \( K_{conn} \) defined in Eqs. 4.1 and 4.2 is used as a Pairwise Master Key defined in IEEE 802.11i.

The EAP authentication consists of authentication message pairs: EAP Request and EAP Response. In IEEE 802.11i, the EAP Request (\( EAP - Req \)) always comes from the AP or an authentication server and EAP Response (\( EAP - Resp \)) comes from the MC as Figure 4.6 shows. To embed my proposed protocols into the EAP framework, I had to extend my protocols with the desired number of dummy messages (marked with ‘-’ in Figure 4.6). The EAP embedded nonce based authentication protocol can be seen in Figure 4.6(a), where the fourth EAP message is a dummy message. In the case of the timestamp based protocol, the first and the fourth EAP messages are dummy messages as it is shown in Figure 4.6(b). Even if the timestamp based protocol
4. FAST AUTHENTICATION METHODS IN WIRELESS MESH NETWORKS

consists of two messages, it is initiated by the MC and not by the AP in accordance with EAP framework. This is the reason that I had to add two additional dummy messages to the original protocol.

The hostapd [Malinen, 2009] on the AP side and wpa_supplicant on the MC side gave an extensible framework for my proof-of-concept EAP implementation. I used the OpenSSL [OpenSSL, 2010] library for the implementation of crypto-primitives. The source code of the implementation is available from the authors upon request.

To measure the total delay of an authentication run, I measured the elapsed time between events sent by wpa_supplicant when authentication starts and successfully ends. I also measure the time consumption of processing an incoming message and generating the response. This measuring process is coded directly into the hostapd and wpa_supplicant application.

Note that I did not consider the delay of 4-way handshake, because it is independent of the authentication method and its delay has been already investigated in other papers (e.g. [Alimian and Aboba, 2004]).

Testbed

Authentication delays were investigated in different scenarios. In each case, the AP was a MikroTik Routerboard 133 (175 MHz MIPS32 CPU, 32 MB memory) with OpenWRT (r11349, kernel v2.6.28.6) installed on it. In order to analyze how the MC’s performance affects the authentication delay, I used three different MCs: 1) high performance (Dell Inspiron 6000 notebook with 1.86 GHz 32 bit CPU), 2) moderate performance (notebook with the CPU running at 800 MHz), and 3) low performance (another MikroTik router with same parameters as the AP has).

I compared my proposal to classical, widely used solutions (e.g. EAP-TLS, EAP-TTLS) with authentication servers (AS). For these cases, I installed hostapd as a stand alone RADIUS [Rigney et al., 2000] server on a PC (with Core2Duo 6400 2.13 GHz CPU, 1 Gb RAM, 32 bit Linux distribution, and kernel v2.6.28). In these scenarios, the AS was connected to the AP with direct link, thus, the roundtrip time between the AS and the MC is minimized.

The type of the wireless card was Atheros AR5414 and Intel 2915 in the case of MikroTik Routerboard and Dell notebook, respectively. The AP and MC communicated through 11g link.

Authentication delay

In this section, I proposed a nonce based (NONCE) and a timestamp (TIME) based authentication scheme with two different certificate sets: one for powerful MCs and another one for constrained MCs (respectively denoted by p and c in the index of the protocol name). I compared these

\[
\text{(a) Nonce based}
\]

\[
\text{(b) Timestamp based}
\]
four authentication proposals to 1) EAP-TTLS [Funk and Blake-Wilson, 2008] with EAP-MD5 [Aboba et al., 2004] inside (TTLS-md5), 2) centralized EAP-TLS [Simon et al., 2008] (TLSa), 3) distributed EAP-TLS (TLSap), 4) EAP-IKEv2 [Tschofenig et al., 2008] (IKEv2), 5) EAP-PAX [Clancy and Arbaugh, 2006] (PAX), and 6) EAP-SAKE [Vanderveen and Soliman, 2006] (SAKE).

Note that EAP-TLS does not require central subscriber management, because it uses only certificates for the authentication and key exchange. Therefore, the TLS connection establishment can be performed at the APs themselves. This is why I differentiated between the centralized and distributed EAP-TLS. In these methods, I used the same certificates and RSA public-private keys as I did in my proposed methods, with pre-generated 1024 bit Diffie-Hellman key parameters.

EAP-PAX and EAP-SAKE are shared-secret based solutions relying on symmetric crypto-primatives, only. In these mechanisms, public key cryptography is used if the MC wants to hide its identity. But this is optional, and I consider scenarios where only symmetric cryptography is used.

I compared the ten authentication scenarios with three different MC devices. The performance benchmark were based on 100 executions of each protocol scenario, and calculating the average authentication delays and their empirical standard deviation of the authentication delays. The performance results can be seen in Figure 4.7. Two subfigures were used to present the results, because the authentication delay in the case of the constrained MC device (shown in Figure 4.7(b)) is at different order of magnitude compared to the delay using the high and moderate performance MC devices which are shown in Figure 4.7(a). On the horizontal axes, different protocols in different scenarios can be seen, while on the vertical axes, the authentication delay are shown. In each scenario, different bars correspond to the measurements made with different MC devices. The whiskers on the top of the bars refers to the empirical standard deviation of the authentication delays. Note that the authentication delay of EAP-TLSap was so long compared to the other measurements in Figure 4.7(a) that I do not show it with complete bar, instead I write explicitly the average value on the top of the reduced bar.

Each of my mechanisms significantly reduced the authentication delay compared to the centralized public key based authentication methods (TTLS-md5, TLSa and IKEv2), where the AS is a powerful entity in contrast to my mechanism, where the AP has limited performance. Furthermore, in the cases of the considered centralized methods, the roundtrip time is minimal, which in a real application, may increase with the latency caused by some wireless hops in the mesh network and with the latency caused by the wired network. The authentication delay in the case of TLSap is even larger, because TLS was not designed for fast connection establishment on constrained devices.

The considered symmetric cryptography based solutions (PAX and SAKE) can complete in around 30-40 ms not taking into consideration the realistic value of the round trip time between the AP and a central authentication server. Note that in the case of high and moderate performance devices, the difference between the symmetric cryptography based solutions and my public key solution is 30-40 ms, but in my certificate based solution there is no further transversal delay. In the case of a constrained device, the delay is considerably higher than in the symmetric cryptography based solutions, but, as I have already described, the centralized solutions have higher vulnerability against DoS attacks.

**Weak key mechanism**

In Figure 4.7(b), the weak key mechanism has significant benefit when the MC has low performance. The overall reduction of the authentication delay is 30% on average in the considered scenario. However, as Figure 4.7(a) shows, the weak key mechanism increases the authentication delay when the MC has high or moderate performance.

To explain this phenomenon, I measured the delay of the processing time of incoming and outgoing messages separately. These are shown in Figure 4.8, where I also compare the authentication delay with and without the weak key mechanism. On the right side of the figures, the bars refer to the AP side, while on the left side, the bars refer to the MC side. In these measurements the transportation delay is not counted, but I show the overall transportation delay at the bottom.
of each figure. These are calculated by getting the difference between the total authentication delay and the sum of all the message processing time. For the sake of simplicity, I shared the transportation delay equally between the AP and the MC.

Note that in Figure 4.8, only those messages are indicated which are related to the original proposal. Thus, processing time of dummy messages sent, because the EAP framework requires it, are counted in the transportation delay. However, the delay of processing dummy messages is negligible.

I considered the timestamp based authentication protocol running by constrained and high performance devices in Figure 4.8(a) and 4.8(b), respectively. The darker color refers to the case when weak keys are used and the lighter color refers to the case when no weak key is used.

In Figure 4.8(a), one can see that the weak key mechanism is very beneficial for the MC, but causes some additional delays at the AP side. The weak key mechanism reduced considerably the delay of the generation of the second message. This process includes the generation of MC’s digital signature with RSA private key. Regarding the verification of this message on AP side, on the one hand, the verification of the digital signature shortens the delay, but, on the other hand, the AP must verify the certificate issued for the weak key and it causes some additional delay. In the generation of the third message, the AP cannot benefit from the usage of the weak key, because the
digital signature generation with DSA can be enhanced by precomputation such that the reduction of the key size does not provide additional significant benefit. Again on the MC side, the usage of the weak key is advantageous, because the verification of the digital signature is reduced. However, the verification of the additional weak key certificate issued by the AP mitigates the positive effect. In the transportation time, there is no significant difference.

Figure 4.8: Message by message comparison of authentication delay of the timestamp based protocol with (constraint) or without weak key mechanism (powerful)

In Figure 4.8(b), one can see the same effects, however, there the MC is powerful, thus, the benefit on the MC side is less significant, and the usage of the weak key mechanism is disadvantageous.

Both Figures 4.8(a) and 4.8(b) show that the weak key mechanism is not beneficial for the AP at all. The reason is that basically all the computationally intensive cryptographical operation was performed by the MC, thus the main improvements can be achieved on that side, but the additional delays caused by the weak key mechanism burden the AP, too.

Even though, only those cases are considered when the weak key mechanism is used on both side or neither side, the mechanism can be used unilaterally, too. To sum up the effect of the weak key mechanism on one side, I can state that it reduces the time consumption of the cryptographic primitives, but also causes additional delays: verification \( t_{cert} \) and transportation of the certificate \( t_{trav} \). From the reduction of the digital signature generation time \( \Delta t_{gen} \) and verification time \( \Delta t_{verif} \) both parties benefit, while the certificate verification delay arise at one party, and the transportation delay depends on the link between the two parties.

Taking these into consideration, in general, the usage of the weak key at one party is beneficial in my proposed authentication scheme if the Eq. (4.3) holds.

\[
\Delta t_{cert}^{(B)} + t_{trav} < \Delta t_{gen}^{(A)} + \Delta t_{verif}^{(B)}
\]
4. FAST AUTHENTICATION METHODS IN WIRELESS MESH NETWORKS

A in upper index refers to the node that generates the certificate and B refers to the other party. \( \Delta t_{\text{op}} \) is the difference between the time consumptions of any operation \( \text{op} \) (\( \text{gen} \) or \( \text{verif} \)) with a long term key (\( t_{\text{op}}(S) \)) and with the weak key (\( t_{\text{op}}(w) \)) as Eq. (4.4) shows.

\[
\Delta t_{\text{op}} = t_{\text{op}}(S) - t_{\text{op}}(w) \tag{4.4}
\]

Using cross-certificates

I also investigated the effect of the cross-certificates. In order to show what is the time consumption of cross-certificates, I considered the timestamp based protocol for powerful mesh clients (i.e., no weak keys are used) with moderate performance and compared the case with and the case without cross-certificates. The result can be seen in Figure 4.9.

![Figure 4.9: Message by message comparison of authentication delay using moderate performance mesh client with or without cross certificates](image)

To generate the first message needs the same amount of time in both cases, because here, it only needs to add the additional certificate to the first message. The AP needs 12 ms on average to verify the cross-certificate. Also, the generation of the second message has some additional delay, because the constrained AP adds its cross-certificate to the message. On the MC side, the verification is very fast as it is a powerful device. The transportation delay increased by 4 ms on average when a cross-certificate is sent by each party.

4.5.7 Evaluation

In this section, I show that my mechanisms satisfy the special requirements relating to the QoS-aware multi-operator environment defined in Section 4.1:

- **Fast authentication method to support user mobility**: My main design principle was to adopt public key cryptography in the considered multi-operator driven mesh network. However, I proved with an implementation and measuring the authentication delay that my proposed schemes reduce authentication delay to an extent that makes seamless handover possible despite the usage of public key cryptography.

- **Mutual authentication**: Both the mesh client and the access point checks the authenticity of the other party.

- **DoS resistance**: The authentication is completely distributed, therefore, an attacker can defeat the access points only one by one. I also minimized the computational load of the APs, especially before the MC becomes authenticated.

- **Compatibility with standards**: I implement my proposals according to the EAP standard, therefore, it suits standard IEEE 802.11i and IEEE 802.11r. Consequently, my authentication
scheme can be used both for inter- and intra-domain handover. Note that my mechanism can coexist with other protocols, and the intra-domain handovers can be handled by other protocols, e.g., that defined in IEEE 802.11r standard.

- **Scalability:** There is no central bottleneck because the authentication and the access control is distributed. A mesh network can be extended by installing valid certificates on the new access points. However, the computational overhead can cause delay if a lot of mesh clients associate to a specific access point at the same time.

- **No single trusted entity:** The access points can authenticate the mesh clients locally. The CRLs can be maintained by each operators’ CA. Therefore, no single trusted entity is required.

- **Connection keys must not reveal long term keys:** The connection keys are based on random numbers generated for the handover and on timestamps, only. Therefore, they do not reveal any long-term key.

- **Independence of connection keys:** The random numbers are generated and the timestamps are read independently of the previous and upcoming connections.

- **Freshness:** The key is controlled by both participants, however, if the key sent by the access point encrypted is compromised, the connection key can be calculated by other parties, too. But only malicious access points send compromised keys and, in that case, they can reveal any not compromised keys, as well.

### 4.6 Future work

In this chapter, two fast authentication methods are proposed. The proposed methods and their investigation can be improved in the following way:

- In the first version of the certificate based authentication methods, as a nonce based authentication method the Blake-Wilson and Menezes protocol was proposed which is a provably secure protocol. The way that I could adapt to my environment causes DoS vulnerability for the access points. I slightly modified the protocol, but I loose the provable secure property. One direction for improving the proposal is to prove formally that the timestamp based and the new nonce based authentication and key agreement protocol is secure.

- As one can read from the results, the highest part of the delay is caused by the decryption of key $K_{AP}$ which is a time consuming operation as it is performed by the access point. My proposal is designed such a way that the key exchange protocol can be modified easily without affecting the authentication part. Novel approaches generating a secret key between two parties by extracting the shared randomness in the wireless fading channel [Zeng et al., 2010] can be utilized to decrease the delay of the whole authentication and key agreement process.

- As mentioned above, the HOKEY and IEEE 802.11r based authentication solution is proposed, but not investigated in this thesis due to the lack of open source HOKEY implementation. The investigation of this solution is considered as a future work too.

### 4.7 Summary

In this chapter, I considered the authentication and access control methods for multi-operator maintained Wireless Mesh Networks with QoS support for mobile users. Based on the Wireless Mesh Networks concept and attacker model described in Section 1.2, I derived the main security requirements. Then, I gave a detailed overview on the state-of-the-art in client authentication and
access control in wireless networks, and I evaluated how the various approaches proposed so far fit the requirements identified for mesh networks.

I found that none of the current standards and related literature satisfy exclusively the requirements on such network. Therefore, 1) I showed how the IEEE 802.11r and HOKEY standards can be combined to suit Multi-WMNs, and 2) I proposed two certificate based authentication protocols that support fast handover in multi-operator maintained wireless mesh networks. For both (nonce and timestamp based) schemes, I proposed two certificate sets: one for powerful mesh clients and one for less powerful mesh clients. In the former set, the computationally intensive operations are shifted to the mesh client, while in the latter certificate set, I proposed the usage of weak keys and short-term certificates for digital signatures.

I created a proof-of-concept implementation, embedded it into the EAP messages, and measured the authentication delay compared to current widely used centralized authentication mechanisms such as EAP-TLS and EAP-TTLS. I found that my mechanism is faster than other certificate based mechanisms even though in my case one party is a constrained access point while the central authentication server is considered to be a powerful PC. I showed that my mechanisms satisfy special requirements relating to the QoS-aware multi-operator environment.

I also investigated how the usage of the weak key mechanism affects the authentication delay by analyzing the processing time of the messages, and I determined when the weak key mechanism is beneficial.
Chapter 5

Misbehaving router detection in Link-state Routing for Wireless Mesh Networks

5.1 Introduction

Ideally, the user should not notice any difference between connecting to the Internet via a wireless mesh network or via a wireless access point that is directly attached to the wired backbone. Hence, providing a high level of QoS is an important requirement in mesh networks. However, the goal of achieving high QoS can be subverted by DoS type attacks, and in particular, by manipulating the basic networking mechanisms such as the routing protocol, the medium access control scheme, the topology control and channel assignment mechanisms, etc. For this reason, it is important to increase the robustness of these basic networking mechanisms. In particular, securing the routing protocol seems to be the most important requirement in this category, because interfering with the routing protocol may affect the entire network, whereas attacks at lower or upper layers seem to have more limited effect.

In the routing layer, I address the problem of detecting misbehaving routers in Wireless Mesh Networks and avoiding them when selecting routes. Misbehaving routers may drop data messages in order to gain an advantage over competitors by dropping messages forwarded on behalf of other operators, or they may lie about their metrics in order to redirect to itself as much traffic as possible, or they may inject fake data messages in order to degrade the QoS level. Since it is very difficult — if not impossible — to defend against misbehaving routers proactively, I essentially propose a reputation system.

The results have been published in [Ács et al., 2010].

This chapter is organized as follows: I give a short overview of the security of routing protocols in Section 5.2. In Section 5.3, I give an overview of the proposed malicious node detection mechanisms. In Section 5.4, I outline my misbehaving node detection mechanism. The system and the attacker models are introduced in Section 5.5. The detailed specification is described in Section 5.6. In Section 5.7, I analyze my mechanism with respect to its performance and speed of adaptivity. Different directions for future work is described in Section 5.8. Finally, I sum up this chapter in Section 5.9.

5.2 Security of routing protocols

In general, routing protocols have a control plane and a data plane. The control plane is responsible for the dissemination of the routing information in the network and for the setup of the appropriate routing tables (or some equivalent routing state). The data plane is responsible for delivering packets to their destinations by routing them using the routing tables.
I differentiate outsider and insider attackers. Outsider attacks both on the control and data plane include deletion of data or control packets by jamming, reordering packets by eavesdropping and replay, as well as injection of fake or modified packets. Cryptographic techniques can be applied to defend against such attacks (except for jamming). The investigation of these mechanisms is out of scope of this thesis, but I assume that the network is protected against outsider attackers. There are many solutions proposed so far (see e.g., [Adjih et al., 2003; Raffo et al., 2004; Hafslund et al., 2004; Buttyán, 2009]).

An insider attacker has all the capabilities of an outsider attacker, and in addition, he can fully control some of the nodes in the network. This means that the attacker can learn the cryptographic secrets of those nodes (if such secrets are used) and he can arbitrarily re-program those nodes. For this reason, insider attacks on the control plane include all deviations from the rules of disseminating, acquiring, and maintaining routing information in the network, while insider attacks on the data plane include dropping, delaying, re-ordering data packets, modifying their content before forwarding them, misrouting them, or any combinations of these misdeeds such that the control packets look genuine (e.g., they can be authenticated by cryptographic means). Insider attacks at the control plane are impossible to detect but their effect on the data plane may be detected, therefore, I focus on detecting insider attacks on the data plane.

Note that the model of insider attackers is realistic, because mesh networks often operate in an environment where physical protection of the nodes is not possible or very costly, and therefore, the nodes can be approached even by an outsider attacker and attacked physically.

Essentially, there are two options to consider as for the type of routing: distance vector routing and link-state routing. These two types of routing have been proposed by the IEEE 802.11s standard too [IEEE 802.11s™/D6.0, 2010]. The main difficulty with distance vector routing is that the routing control packets contain untraceable aggregated routing metric values that are legitimately manipulated by the nodes that process those control packets. I consider this as an important disadvantage of distance vector routing, therefore I choose the link-state routing approach.

I define informally a misbehaving link as a link whose behavior is not consistent with the routing information disseminated or acquired by the protocols operating at the control plane. Note that such inconsistency may result not only from misbehavior at the data plane, but also from the dissemination of incorrect routing information at the control plane. I do not intend to make a distinction between these two cases, I simply want to detect the misbehaving routers at the data plane. In addition, it should also be possible to inform the non-corrupted routers about the identified misbehaving links such that they can avoid them when selecting routes later on.

5.3 State-of-the-art

Approaches for misbehavior detection at the data plane of routing fall into three families: (1) acknowledgement schemes, (2) traffic monitoring, and (3) neighbor monitoring.

Acknowledgement schemes These schemes use acknowledgements to detect data packet dropping on a route. An interesting adaptive acknowledgement scheme for detecting misbehavior at the data plane in ad hoc networks is proposed in [Awerbuch et al., 2007]. This approach requires that the nodes use source routing, and therefore, the source knows the entire route to the destination. The idea is the following: The destination is required to return an acknowledgement for every packet that it receives successfully. Based on these acknowledgements, the source keeps track of the loss rate in a time window of a given size. If the loss rate exceeds a threshold, the source starts a binary search on the route to identify the misbehaving node, or more precisely, the link that causes the delivery failure of the packets. For this, the source adaptively specifies a list of intermediate nodes in the subsequent packets that should also return an acknowledgment for the packets that they successfully processed. These nodes are called probe nodes. First, one probe node is selected in the middle of the path between the source and the destination. If the acknowledgements arrive from this node but not from the destination, then the bad link must be between the probe node and the destination. Otherwise, if the acknowledgements do not arrive
from the probe node either, then the bad link must be between the source and the probe node. Once the sub-path that contains the bad link is identified, a new probe node is specified in the middle of that sub-path. This procedure is continued until the sub-path that contains the bad link is narrowed down to a single link, which must be the bad link. The misbehaving node can be either end of the identified bad link.

The main disadvantage of this method seems to be its extended detection time when faced with colluding misbehaving nodes that are placed strategically on a route. For example, it needs a lot of time to locate a path segment of colluding misbehaving nodes that overlaps at least two binary probing intervals.

In [Herzberg and Kutten, 2000], a theoretical framework is proposed for constructing acknowledgement schemes that try to minimize the time needed for detecting misbehaving routers in wired networks. The main premise of this approach is that the actual delivery time of a message over a link is usually much smaller than the a priori known upper bound on that delivery time. By taking advantage of this observation, the authors develop an abstract model for various time-optimal or communication-optimal acknowledgement schemes that detect and locate any misbehaving link or path segment.

Another acknowledgement scheme is called 2ACK [Liu et al., 2007], because each router on a path sends acknowledgements to its two-hop neighbor on the path in the reverse direction (i.e., opposite to the direction of data forwarding). Only a fraction of the received data packets are acknowledged to reduce overhead. The main disadvantage here is that this approach cannot detect three or more colluding misbehaving routers in a row.

Traffic monitoring These approaches are based on the Conservation of Flow principle, which says that if a router behaves correctly, then the amount of transit traffic entering in the router should be equal to the amount of transit traffic leaving that router. In order to verify that this principle is respected in the network, each router counts data packets of different types, periodically exchanges its counters with other routers, and check if the counters are consistent with each other and with the Conservation of Flow principle. This approach has a low overhead and can be effective if implemented correctly. However, in its basic form, it does not detect packet modifications, reordering, and delay [Hughes et al., 2000]. Several specific misbehavior detection mechanisms based on this traffic monitoring approach have been proposed for wired networks, including WATCHERS [Bradley et al., 1998] and FATIH [Mizrak et al., 2006], and for wireless networks [Gonzalez et al., 2008].

Similarly to my solution, in [Tourolle et al., 2007], the authors introduced a reputation system, where the nodes are evaluated by gateways. The reputation value of each node in a route increases equally if a message arrives to the gateway and decreases if not. The detection of the packet loss is based on the TCP connection control mechanism. Therefore, this solution does not require any further control messages to evaluate the routers. However, the reputation value of all the routers decreases if one router misbehaves. In contrast to this, my solution try to pin-point the misbehaving routers one-by-one. Another disadvantage of this approach is that the evaluation is based on only TCP message flows.

Neighbor monitoring These approaches exploit the broadcast nature of the wireless communication medium, by requiring that routers continuously monitor the activities of their neighbors and try to detect misbehavior. More specifically, a correctly behaving node can detect that one of its neighbors has received a packet that it should forward, but it does not. This kind of monitoring can be implemented by putting the network interface of the nodes in promiscuous mode (most interface cards allow this) and by listening to everything in the wireless channel. If a node does not overhear the retransmission of a packet by its neighbor, then that neighbor can be suspected to misbehave. This sounds simple, but in practice, there may be many issues that make this approach difficult to use. For instance, if the nodes use multiple channels and radios, then they may not hear their neighbors retransmitting the packets. Similar problem may arise, when the nodes use power control to adaptively adjust their transmission range. There are also issues related to the hidden
terminal problem and to skipping re-transmissions after an unsuccessful transmission. Watchdog and Pathrater [Marti et al., 2000] are two mechanisms that together implement a misbehavior detection and mitigation tool based on neighbor monitoring. Watchdog is in charge of continuously monitoring neighbors and trying to identify misbehaving nodes that do not forward data packets that they should forward. Pathrater is used to select routes that likely avoid those misbehaving nodes. The operation of Watchdog is based on listening in the promiscuous mode and trying to catch the transmission of the data packet by the neighbor to which it was forwarded. Pathrater assumes that each node maintains a rating in the interval [0, 1] for all the other nodes it knows in the network. Then, the reliability of a route is quantified by the source of a data packet by averaging the ratings of the nodes in that route. The nodes prefer routes with a higher average rating.

5.4 My approach

The misbehavior detection can be carried out by evaluating the links or the nodes. On one hand, it is easier to evaluate links because the nodes disseminate information about the links, and the stated link parameters can be compared to the observed link parameters. On the other hand, the misbehavior may be performed by a Byzantine attacker which can quickly change the attacked link. Therefore, I evaluate nodes, not links.

My goal is to identify misbehaving routers at the data plane, and provide some feedback to the control plane that helps to avoid the identified misbehaving routers during the path selection procedure. I assume that gateway nodes are better protected physically than regular mesh nodes, and therefore, I assume that they behave correctly. As one end of each path is always a gateway, I let the gateways control the misbehavior detection mechanism. I assume that the mesh clients do not participate in the mesh routing protocol, which implies that the other end of each path is an access point. As it is very difficult to identify misbehaving end-points at the routing layer, I limit ourselves to the detection of misbehaving intermediate mesh routers.

My misbehaving router detection protocol consists of three phases. In the first phase, called traffic validation, each gateway collects information about the forwarding behavior of the routers on the paths belonging to the given gateway. In the second phase, called router evaluation, the gateways attempt to identify suspicious routers based on the traffic information collected in the previous phase. As a result of the router evaluation phase, the gateways compute Node Trust Values, and disseminate those within the network. Finally, in the third phase, called reaction, the routers select new routes by taking into account the Node Trust Values of the other routers.

In order to support traffic validation, I require each node (including access point, routers and gateway) only to maintain a counter for each path it is part of to count the number of data packets that it forwards on a given path. I assume that each data packet has a routing header that contains a path identifier and message authentication codes. Thus, intermediate routers can verify the data packets and they count only intact packets. The packet counters that belong to a given path are requested by the gateway in a regular manner (e.g., periodically, based on sequence number of data messages, or when a specific control message is received), and the routers on the path report them to the gateway.

As misbehaving routers may report fake counter values, the gateway does not use the reported counters directly in the computation of the Node Trust Values. Instead, the gateway considers different explanations for a set of received counter values. In each explanation, each intermediate router is either accused for misbehavior or considered honest, thus explanations are essentially binary vectors. An explanation is valid if two simple principles hold for it: 1) if there is a difference between the counters of two neighboring routers, at least one of them must be accused, and 2) if the counters are equal for two nodes and none of them is accused, no other nodes must be accused between them. The Node Trust Value of a given router is computed as a weighted sum of its accusations, where explanations that contain fewer accusations have higher weights. The computed Node Trust Values are fed back in the system using acknowledgement scheme.

A router may receive multiple different Node Trust Values for a given router from different
5. MISBEHAVING ROUTER DETECTION FOR WIRELESS MESH NETWORKS

gateways. The router aggregates those trust values by either averaging them or taking the minimum of the received values. The resulting aggregate trust value computed for a router $i$ is then used as follows: the router excludes router $i$ from its topology view with probability proportional to the aggregate trust value of router $i$ and establishes new paths using this reduced topology view. Thus, less trusted routers are less likely to be considered as potential intermediate routers on the selected paths. Note that the approach where nodes are excluded if the Node Trust Value goes below a threshold has the disadvantage that excluded nodes do not have chance to increase their reputation value.

In order to go into details regarding to the router evaluation and reaction phase, the system and the attacker model is introduced in the next section.

5.5 System and attacker model

5.5.1 System model

The mesh nodes are placed uniformly at random in an arbitrary two-dimensional field. All the mesh nodes are equipped with wireless interface(s) with the same radio range. Two mesh nodes are neighbors if they are within each other’s radio range. Each node has a wired connection to the Internet (i.e., play the role of the gateway) with probability $\varphi$. Every node is malicious with probability $\varsigma$ except for the gateways which are assumed to be trusted.

For the sake of simplicity, I assume that each link has high quality, and a MAC protocol assures that the messages are successfully delivered between neighboring nodes. Thus, the only reason for dropping a data packet is the malicious behavior of some routers in the data plane.

I assume that a router is allowed to reject to participate in a path. Otherwise, the routers are motivated to decrease their reputation value in order to force to be avoided as a router. This is required to be able to provide the promised QoS level for mesh clients when the router plays the role of access point.

Without loss of generality, I consider only one direction of the traffic. In particular, the nodes send their counters which refer to the upstream traffic, only. The mechanism described below can be adapted to the analysis of the downstream traffic analogously.

I assume that all the traffic counter reports arrive to the gateway. This is necessary, otherwise the gateway is not able to evaluate the routers. In a real implementation, an acknowledgement scheme can be used to be able to detect the loss of a counter report. In the case of unsuccessful delivery, the router can re-send the counter report by flooding the network. It may happen that malicious nodes form a vertex cut, in which case they can prevent the reception of the traffic counter reports from honest nodes and these honest nodes are temporarily excluded from the network. The routers which do not receive any control packets from the gateway, go into idle state, and do not participate in the routing until they reach the gateway again.

Note that in mesh networks, a node does not necessarily need to learn any information about distant nodes, it only has to reach some of the gateways that are close to the node. In order to limit the number of control messages flooded in the network, I consider relatively short TTL (Time-To-Live) values for control message. Therefore, a node only learns a part of the whole network, and the nodes will have (typically) different Views of the network.

5.5.2 Attacker model

As I have already described, I do not distinguish if a malicious router reports better link states than it has in reality or it simply drops the data packets. But I assume that an attacker wants to redirect as much traffic as possible by better link reports. Note that if a malicious router reports a link quality that is lower than the actual quality of the link then the access points will choose paths that bypass the malicious routers, and therefore, this behavior is not beneficial for the attacker. Therefore, the malicious router is modeled by dropping each data packet with probability $\vartheta$. 
The upstream counter $cnt_i$ of router $i$ is meant to count the number of data packets that traverse router $i$. However, misbehaving routers may not correctly set their counters. Let us consider a simple case when a malicious router $i$ is placed between two honest nodes. The malicious router has three options when it sets its counter that it sends to the gateway:

- **The attacker sets its counter to the number of incoming data packets** $cnt_{in}^i$ (i.e., $cnt_i = cnt_{in}^i$). In this case, the gateway realizes that on the link before the malicious router, there is no lost data packet as $cnt_i = cnt_{in}^i$. But on the next link, the difference is $cnt_{in}^{i+1} - cnt_i$. It is impossible to decide at the gateway side if node $i$ indeed forwarded all the data packets and node $i+1$ dropped them, or node $i$ dropped them, and node $i+1$ received only $cnt_{in}^{i+1}$ data packets. Therefore, in this case, nodes $i$ and $i+1$ both should become suspicious.

- **When the attacker sets its counter to the number of outgoing data packets** $cnt_{out}^i$ (i.e., $cnt_i = cnt_{out}^i$), again the gateway finds two suspicious nodes: node $i-1$ and $i$. It is indistinguishable from the value of the counters if node $i-1$ dropped the $cnt_{in}^i - cnt_{in}^{i-1}$ data packet and node $i$ forwarded the rest honestly, or node $i$ dropped them.

- **The attacker can also choose randomly a number such that** $cnt_{in}^i > cnt_i > cnt_{out}^i$. I will show that this case is the least beneficial for the attacker in my router evaluation mechanism. Therefore, I only consider the first two cases.

When it is requested by the source node on the route (the access point or the gateway depending on the direction of the route), a malicious router sends the value of incoming counter as the traffic counter value with probability $\xi$ and sends the value of outgoing counter with probability $1 - \xi$. I also investigate extreme scenarios when $\xi = 0$ and $\xi = 1$.

### 5.6 Node Trust Value

#### 5.6.1 Calculation of Node Trust Value on each route

As I have described, the gateways evaluate the node behavior on each route separately. For this, the gateway inspects the counter reports received from the routers. If every router behaves correctly then each counter value should be the same, as no packet is dropped on the route. On the other hand, if there are misbehaving routers on the route, then there must be a link where the counter values received from the two ends of the link are different. There may be different explanations supporting a given set of counter values where an explanation contains an assumption for each router regarding its correctness. More specifically, an explanation $\exp$ is a vector, where the $i$th element of the vector is 0 if the $i$th router in the route is assumed to be misbehaving — suspicious or accused in short —, otherwise, the $i$th element is 1.

An explanation is valid if all of the following statements hold:

- If there are data packets lost between node $i$ and $i+1$, at least one of them is accused.
- If node $i$ and node $j$ are not malicious in the given explanation, and there is no data packet loss between them, none of the nodes between $i$ and $j$ are accused.

Weights are assigned to each explanation of a counter report. I consider two kinds of calculation of the weights, both depends on the number of suspicious nodes in the explanation. Let us denote the number of suspicious nodes in explanation $\exp$ by $\|\exp\|$ and the number of all routers in the given path by $\|\exp\|$. The two different weight function $w_1(\cdot)$ and $w_2(\cdot)$ defined in Eqs. (5.1) and (5.2), respectively.

\[
w_1(\exp) = q^{\|\exp\|} \cdot (1 - q)^{\|\exp\| - \|\exp\|}, 0 < q \leq 1 \tag{5.1}
\]

\[
w_2(\exp) = \begin{cases} 1 & \text{if } \|\exp\| = \min_{\exp'}(\|\exp'\|) \\ 0 & \text{else} \end{cases} \tag{5.2}
\]
5. MISBEHAVING ROUTER DETECTION FOR WIRELESS MESH NETWORKS

One extreme explanation, if some data packets are lost, is that all of the nodes are suspicious. This is possible, but not too realistic. In Eq. (5.1), I defined a function that assigns higher weight to those explanations which include fewer suspicious nodes as usually (to be more precise, when the probability that a node is malicious is low) these explanations have a higher probability. In Eq. (5.1), \( q \) denotes the probability of a node becoming malicious. In my analysis for the sake of simplicity, I assume that this probability can be guessed accurately, therefore, \( q = \varsigma \).

Using Eq. (5.2) as a weight function, I consider only those explanations which include the lowest number of suspicious nodes, the other explanations are discarded.

Given the set of possible explanations \( \exp \) for a given set of counter reports, a gateway \( g \) which is one end of the route \( r \) calculates at time \( t \) the Node Trust Value of router \( i \) denoted by \( \eta_{r,g}^{(t)} \) in the following way:

\[
\eta_{r,g}^{(t)} = \sum_{\forall \exp} \frac{w(|\exp_e|)}{\sum_{\forall \exp} w(|\exp_e|)} \cdot \exp_e(i) \tag{5.3}
\]

where each explanation \( \exp_e(i) \) is weighted using the normalized value of one of the previously described weight function.

The properties of the \( \eta_{r,g}^{(t)} \) are the followings:

- The \( \eta_{r,g}^{(t)} \) is always in the interval \([0, 1] \)
- If router \( i \) is suspicious in each possible explanation, \( \eta_{r,g}^{(t)} \) equals to 0
- If router \( i \) is not suspicious in any of the possible explanation, \( \eta_{r,g}^{(t)} \) equals to 1

I have stated that it is not beneficial for a malicious router to send a counter value between the number of incoming and outgoing data packets. The reason is the following. If the router choose a number in between, the explanation where the malicious node is the only suspicious node involves the least number of suspicious nodes. Therefore, both weighting methods will render higher weight to this explanation than to the others.

5.6.2 Aggregation of Node Trust Values

Note that a gateway may evaluate routers through multiple routes, and access points may receive multiple Node Trust Values from multiple gateways. Therefore, a mechanism for aggregation of Node Trust Values is required.

Note that access points may receive Node Trust Values from different gateways (as the subnet-work that they see can differ due to the short TTL values of control messages). Therefore, they may calculate different NTVs when they aggregate them. Since the path selection is performed by the access points independently from each other, it does not cause any issues.

Each \( \eta_{r,g}^{(t)} \) are utilized using an \( n \) long window. There is a window for each router in the View of the gateway. These values may be calculated from different routes \( r_k \) or the same route but from different time \( t_l \) using function \( f \):

\[
\eta_{r,g}^{(gw)} = f(\eta_{r_1,g}, \eta_{r_2,g}, \ldots, \eta_{r_{m},g}) \tag{5.4}
\]

When access point \( a \) receives multiple \( \eta_{r_g}^{(gw)} \) from different gateways \( g_k \), it only stores the latest value from each gateway. The Node Trust Value that the access point calculates is denoted by \( \eta_{a,i}^{(ap)} \) and calculated using the function \( f \):

\[
\eta_{a,i}^{(ap)} = f(\eta_{a,g_1}^{(gw)}, \eta_{a,g_2}^{(gw)}, \ldots, \eta_{a,g_m}^{(gw)}) \tag{5.5}
\]

where \( m \) is the number of gateways that have sent Node Trust Value about router \( i \).

I investigate the minimum and the average function as \( f \) in Section 5.7.
5. MISBEHAVING ROUTER DETECTION FOR WIRELESS MESH NETWORKS

5.6.3 Utilizing the Node Trust Value aggregated by the access points

When access point $i$ has to establish a path to a gateway, it uses the $\eta_{i,j}^{ap}$ to avoid routes that include malicious nodes. One of my objectives is to propose a mechanism which utilizes the aggregated Node Trust Values, but it does not require any modification neither on the link-state routing protocol, nor on its route selection mechanism in order to consider the QoS and the trust values simultaneously.

I achieve the requirement described above in such a way that instead of considering each router in the View of the access point, I determine a subview which the route selection runs on. Access point $i$ includes router $j$ into the subview with probability $\eta_{i,j}^{ap}$.

Note that with this approach, nodes in the subview may form a graph that is not connected, therefore, there is no guarantee that the access point can find any route to any gateway. If it happens, new subview generation is required. Nevertheless, in order to ensure that the procedure terminates, I define a threshold, which is initially 1, and the threshold decreases in each unsuccessful subview generation by $\nu$. All the routers $i$ for which $\eta_{i,a}^{ap} > 1 - r \cdot \nu$ are included in the subview ($r$ is number of unsuccessful trials).

This method assures on the one hand that an access point finds a route to a gateway within at most $\nu^{-1}$ try. In the worst case all the nodes are included, and the method works as if there were no any defense mechanism. On the other hand the routers which seem to be malicious have a chance to be included in the route. Therefore, they can improve their Node Trust Value if they are indeed honest or start to behave honestly.

5.7 Performance analysis

In order to investigate the effectiveness of the proposed mechanisms, I run simulations.

5.7.1 Simulations

200 mesh nodes are placed uniformly at random in a two-dimensional $10 \times 10$ unit field. The radio range is 1 unit. A node is gateway with probability $\varphi$, which is 0.1 in the considered scenarios. Only those scenarios are considered where each node can reach at least one gateway. If the gateways could be reached, but they are out of the View of a node, the depth of the View is enlarged which is 4 by default.

It is not necessary that there exists any route between two arbitrary mesh nodes. However, I require that any mesh node must find a route to at least one gateway. If a scenario does not fulfill this requirement I generate a new one.

The simulation is divided into rounds. In each round, a randomly chosen source builds a route to a gateway, and sends 100 data packets. After every 10th data packets, each router sends its upstream counter to the gateway. After each report, a gateway $g$ calculates $\eta_{i,g}^{r(t)}$ for each router $i$, and updates its table. The table at index $i$ stores the last 30 $\eta_{i,g}^{r(t)}$ for each $r$ and $t$. Finally, it calculates the $\eta_{i,g}^{(gw)}$, and disseminates among the nodes that are in its View.

When I specified my secure routing mechanism, I stated that there are pre-established routes between access points and gateways. For the sake of simplicity, in the simulations, the access points choose from all possible routes in the subview.

I divided the whole simulations into three phases. The first phase is the bootstrap phase. The Node Trust Value of each router is initially 1, i.e. they are fully trusted, however some of them are malicious. I determined experimentally that the Node Trust Values reach their steady-state values within 2500 rounds, and therefore, I set the length of the first part of the simulation to this number of rounds. Similarly, the second phase lasted for 2500 rounds, too. In the second phase, the subset of the routers are still malicious, but here the access points have a clearer view of the network. In this phase, I collected statistics from which I investigated the properties of my mechanism (Figure 5.2, 5.3, and 5.4). In the last long phase which lasts for 5000 rounds, all...
the malicious nodes behave honestly. With this, I could investigate the speed of adaptivity of my mechanism.

The source of each route is an access point. Any node can play the role of the access point, but I consider only 2-hop or longer routes, otherwise none of the participants can behave maliciously due to my system and attacker model. The access points choose uniformly at random from each possible shortest path that leads to any gateways on the currently generated subview.

The route selection algorithm is based on the hop count. In each round, the access point chooses the shortest path. Note when some routers are discarded, the router select the shortest path on the subview which may increases the length of the routes. Therefore, I also investigate the path length. I utilized the Dijkstra algorithm to determine the shortest paths. The access points choose uniformly at random from each possible shortest path.

Recall that different access points may calculate different Node Trust Values as they may aggregate different Node Trust Values received from different gateways. Thus, it is difficult to plot their value. For the sake of simplicity, when I show the Node Trust Value of a router, I consider the \( \eta_{i}^{(ap)} \) which is calculated by the node that the Node Trust Value is about. Note that it is the Node Trust Value where the router includes all the gateways that evaluate the router. In the following, I refer to these values simply as NTV. In Table 5.1, I summarized the fixed parameter values of the simulations.

| Field size | 10 x 10 |
| Radio range | 1 |
| Number of nodes | 200 |
| Probability of being gateway (\( \varphi \)) | 0.1 |
| Number of simulations | 10 |
| Simulation length | 10000 |
| Number of messages in each round | 100 |
| Size of View | 4 |
| Size of window at gateways (\( n \)) | 30 |
| Node Trust Value threshold step (\( \nu \)) | 0.05 |
| Initial Node Trust Value | 1 |

I considered different values for the probability \( \varsigma \) of being malicious, for the probability \( \vartheta \) of dropping a data packet, and for the probability \( \xi \) of reporting the counter of the incoming data packets to the gateway. I run simulations with the values shown in Table 5.2. A default scenario is described with the parameters indicated by bold text in the same table. As different scenarios do not show significant or unexpected changes, only the default scenario is analyzed in detail.

| Probability of being malicious (\( \varsigma \)) | 0.05 | 0.2 | 0.5 |
| Probability of dropping a packet (\( \vartheta \)) | 0.2 | 0.5 | 1 |
| Prob. of reporting \( cnt_{in} \) (\( \xi \)) | 0 | 0.5 | 1 |

In Figure 5.1, I show a sample scenario generated with default parameters. The nodes which are neighbors are connected with a line. The gateways are denoted by large white circles \( \Box \), while the malicious nodes are denoted by large white triangles \( \bigtriangleup \). All other nodes are represented by small black circles \( \bullet \).

5.7.2 Results

Recall that different access points may calculate different Node Trust Values. In the figures I show the average Node Trust Value that includes all the gateways that have evaluated the router. Recall, I refer to these values simply as NTV.
Figure 5.1: **Sample scenario** Gateways are denoted by \(\bigcirc\), malicious nodes are denoted by \(\Delta\), and regular mesh routers are denoted by \(\bigstar\).

In Figure 5.2, the NTV of three different groups can be seen with the 0.95 confidence intervals. The routers are categorized into three different groups: 1) malicious routers, 2) honest routers which are neighbors of malicious routers, and 3) other honest routers. I analyzed the latter two groups separately because the malicious routers can degrade the Node Trust Value of the neighboring nodes when the gateway evaluate the received upstream counters. At each group, four bars can be seen. The bars refer to different parameters of the malicious node detection mechanism. The **all** and **least** indicate the usage of Eq. (5.1) and (5.2), respectively. The NTV is aggregated using the function minimum or average when the bar is indicated with **min** or **avg**, respectively.

As Figure 5.2 shows, the NTV of the honest nodes is maximal. In particular, the honest nodes are usually included in the subview which the route is selected from. In contrast, the average Node Trust Value of the malicious nodes is almost zero when the minimum function is used for the aggregation. This means that the malicious nodes are bypassed with high probability. If the Node Trust Values are aggregated by calculating the average function, the values are higher, but the difference is still significant between the average NTV of the honest and malicious nodes.
Considering the neighbors of the malicious nodes, the NTVs are relatively high, but as I expected, significantly lower than of the other honest nodes.

Note that average NTVs do not show significant differences when Eq. (5.1) or (5.2) is used. In some scenarios (e.g., when $\varsigma = 0.5$), with the former one, the NTVs of the malicious nodes is less, but also the NTVs of the neighbors of them and the honest nodes is less. Nevertheless, the probability of a node being malicious is a priori known and exploited in Eq. (5.1), which is not a realistic assumption. The investigation of the right parameter of $q$ is considered as a future work.

![Figure 5.3: Average numbers of dropped data packets with 0.95 confidence intervals](image)

In Figure 5.3, the average number of dropped data packets are shown with 0.95 confidence intervals using different parameters of misbehavior node detection mechanism. These results are compared to the case when no defense mechanism is used at all. As one can see, the number of data packet drop is reduced with my mechanism considerably. It worked somewhat better with the minimum aggregation function than with the average function, which comes from the fact that the malicious nodes are excluded from the subviews with higher probability.

I also investigate the cost of avoiding malicious nodes by my mechanism. My simple QoS metric is the hop number. Thus, average length and the 0.95 confidence interval of the number of hops is shown in Figure 5.4. I indicate only above 2 hops, because it was the minimum hop number in the considered scenarios. As one can see, the length of routes does not increase significantly with my mechanism. This comes from the fact that in many cases, the access points could choose alternative routes which had the same length as the route that contained malicious routers, too.

![Figure 5.4: Average lengths of the routes with 0.95 confidence intervals](image)

In Figure 5.5, the NTVs are grouped into the three group and their average value are plotted against the time. There, I investigate how fast my mechanism adapts to the case when the nodes become malicious or they are repaired. Recall that initially the routers are fully trusted and in the first part of the simulation (first 5000 rounds), some nodes are malicious, while in the last part (last 5000 rounds), the malicious nodes are repaired and do not drop any packets.
In Figure 5.5(a), the process of changing of the NTV is plotted when the Node Trust Values are aggregated with average function, while in Figure 5.5(b), the aggregation function is the minimum function. In both cases, the Node Trust Values are calculated with Eq. (5.2).

![Graph (a) Average Node Trust Value Adaptation](image-a.png)

![Graph (b) Minimum Node Trust Value Adaptation](image-b.png)

As it is emphasized in Figure 5.5(a), the 90% of the final NTV (at 5000\textsuperscript{th} round) is reached after 670 rounds. Recall that one route is evaluated in each round. In both figures, the average NTVs reach their value in round 5000 at the same speed. In contrast to this, in the second part, after the nodes are repaired, the average NTV of the misbehaving nodes return faster to 1 when the aggregation function is the average. The reasons are the following. Firstly, with the average function, the NTVs of the malicious nodes are higher, therefore, they are selected in the routes with higher probability than with minimum function. Therefore, they have more chance to increase the low NTVs. Furthermore, when a repaired router \( i \) obtains high \( \eta_{r(t)}^i \), it falls out low \( \eta_{r(t)}^i \) from the window maintained by gateway \( g \). Therefore, the router certainly increases its NTV. In contrast to this, in the case of minimum function, the lowest \( \eta_{r(t)}^i \) may be not fallen out. Therefore, the router will have again low chance to increase its NTV.

This is always a trade-off in the reputation systems. If a node with low reputation value has many chances to correct, it has the opportunity to abuse the system. On the other hand, if a node with low reputation value is excluded from the system, it has little chance to be involved again.

I did not investigate the overhead of my mechanism in the simulator, but I think that the overhead is insignificant. In each report period each node has to send the counter value to the gateway (a node may flood the network only if its counter value did not arrive to the gateway) and the gateway floods the updated Node Trust Values in its View.
5.8 Future work

In this chapter, I have shown that the idea of explanation based reputation system is promising. However, the elaboration of both the algorithm and the investigation can be improved in some ways:

- In the current proposal, the functions that aggregate the NTV values are very simple and these can not exploit all the possibilities of the explanation based approach. More complex aggregation functions could identify better the nodes who misbehave on different paths.

- Currently, my solution can only identify routers which drop messages. However, in real applications more sophisticated QoS levels are provided, which also requires my approach to adapt to.

- A more realistic simulation environment such as OMNeT++ [Böthe and Varga, 2011] or ns3 [Lacage and Dowell, 2011] would give better evaluation of the proposal. In such simulators, it is easy to simulate the non-intentional link failures too. This would make possible to relax a related assumption, and make my solution more robust. Also realistic simulators are able to evaluate more sophisticated QoS values.

5.9 Summary

In this chapter, I proposed a novel misbehaving router detection mechanism for link-state routing protocols in wireless mesh networks. My approach is based on calculating reputation values for each router in the network. The reputation value is based on the counter that routers regularly send, and the counter counts the number of forwarded packets. After each report, the gateway takes into consideration all possible explanations — who can be malicious — that explain the packet loss. I designed the reputation value utilizing mechanism in such a way that it does not make any restriction on the QoS aware route selection mechanism.

I showed that my misbehaving node detection mechanism bounds low trust value to misbehaving nodes, while the node trust value of the honest nodes remained high. I found that with my mechanism, the number of dropped data packets was much lower compared to the case when no defense was applied. Furthermore, the length of the selected path did not increase considerably.

In order to show that my mechanism works in practical environment I implemented my mechanism as an extension to olsrd (see www.olsr.org) and compiled for OpenWRT, which is a Linux distribution for embedded devices such as mesh routers.
In this thesis, two instances of multi-hop wireless networks are considered: 1) Wireless Mesh Networks, and 2) Delay Tolerant Networks. Different issues are investigated in each issue. In this chapter, the application of new results presented in this thesis is described.

**Delay Tolerant Networks.** Regarding the Delay Tolerant Networks, I imagined a scenario where local information needs to be distributed to a set of nearby destinations based on their interest in the information such as the touristic example shown in Section 1.1.1. These networks consist of handheld devices belonging to individual mobile users.

Recall that in the touristic example, instead of setting up an on-line bulletin board, where the tourists have to pay both for the usage of bulletin board service and for accessing the network, a city-wide Delay Tolerant Network can provide a very cheap alternative solution. In this solution, touristic information can be distributed in a store-carry-and-forward manner by using Bluetooth capable devices (e.g., mobile phones, PDAs) and by exploiting the mobility of the tourists themselves.

In another example, the infrastructure can be substituted by the Delay Tolerant Network approach in rural areas too, where the providing Internet service is not profitable such as in developed countries. DakNet Project [Pentland et al., 2004] make the Internet available without last-mile broadband infrastructure in rural villages in India. In each village, a kiosk was built with short-range wireless radio which forwarded the requests and downloaded the response from the buses, motorcycles equipped with mobile access points. Mobile access points communicated with fix access points in a near town. DakNet supported messaging (e.g., e-mail, audio/video messaging), distribution of information (e.g., bulletin board, news, public health announcements), and collection of information (e.g., voting, environmental sensor information).

In Vehicular Ad hoc Networks (VANET), the cars communicate with each other in order 1) to provide higher safety and/or higher permeability for the traffic, or 2) to entertain the passengers. In order to satisfy some special requirements of VANET, current design principles require roadside infrastructure. However, due to the high price of the roadside infrastructure installation, in the sparsely built-up areas, the vehicles can only communicate with each others with DTN approach. Considering the entertainment applications in VANET, usually it is not worth to build up any infrastructure, therefore, the vehicles will probably communicate with each other according to DTN approach, too.

In the typical DTN scenarios as described above, the end users are responsible for the data forwarding due to the lack of the infrastructure. DTN networks compared to infrastructure based networks promise cheap but not reliable data forwarding. Reliability can be increased by motivating the users to forward each others’ messages. Considering an extreme situation, if none of the users forward messages on behalves of other users, messages are destined only when the source and the destination nodes meet each other. In practice considering the above described examples, it could
mean that 1) tourists learn that a museum is closed when they go to the museum, 2) the kiosk receives a request to forward only after the bus equipped with mobile access point going to city has left, and 3) a car receives information about an accident when it is close to the accident instead of at a fork in the road where the accident can be bypassed easily on an alternative route. The barter based solution proposed in Chapter 2 encourages users to disseminate messages on behalves of other nodes making reasonable to supplement infrastructures with DTN approach.

The traceability also can be an issue in DTN networks. Due to the fact that the end users store and forward the message while they are moving, the users could become traceable if the message forwarding protocol is not designed carefully. Considering the tourist example, if the tourists are traceable, they can be targeted by advertisements based on the information what place they visit. Also in rural areas and in VANET networks, the privacy should not be violated. Anonymous communication has been investigated in the context of mobile ad hoc networks, too. But there is a DTN specific problem due to the store-carry-and-forward data dissemination manner. In particular, users can be profiled based on the information that they store and they want to download. In Chapter 3, the Hide-and-Lie solution is proposed in order to prevent attackers to trace the nodes who apply the solution.

The barter and Hide-and-Lie proposals and their investigation was applied to BIONETS (BI-Ologically inspired NETwork and Services) EU project. The FP6 project aimed to develop an infrastructureless network providing evolutionary services that adapt to the users’ needs automatically at the service layer. BIONETS networks also have to handle a huge number of nodes with wide heterogeneity in node capabilities at the network layer. The project started in 2006 and successfully ended in February of 2010. More information can be found at http://www.bionets.eu.

The investigation of both barter and Hide-and-Lie mechanisms were delivered to BIONETS as technical reports. Furthermore, the barter mechanism was integrated to the BIONETS Simulator. BIONETS Simulator is an OMNeT++ based [Böjthe and Varga, 2011] simulator through which the results of the whole project were presented in an integrated form.

**Wireless Mesh Networks.** Wireless Mesh Networks provide last mile broadband access for mobile users who may run QoS aware applications. Maintaining them by multiple operators could be advantageous even if the operators compete with each other. Furthermore, the multi-channel approach with multiple wireless interfaces could have high gain for the operators.

As a remainder, some advantages of multi-operator maintenance are listed here: 1) due to the shared costs of installation, the coverage can be increased at lower cost per operator, 2) the spectrum can be utilized better, because some control mechanisms can be applied in order to decrease the packet collision which can be hardly implemented when the overlapping networks are independent. Recall when the network is connected using multiple channels, the bandwidth may increase, but the connectivity of the network may decrease since the routers that uses different channels can not communicate even if they are in each others coverage. The multiple interface approach can mitigate this drawback.

Such networks can be utilized by mobile business users. Mobile business users require access to services and applications for work while they are traveling. When the mobile business users travel on vehicles (e.g., train, bus), they can use their laptop, smart-phones, PDA, etc. for accessing a full range of corporate services. They can also run real time multimedia applications (e.g., video). When the mobile business users are pedestrians, they can use their smart-phones making VoIP calls or accessing any kind of streaming audio. The employee effectiveness can be improved and the companies can take advantage of these applications if the network access is cheap as the Wireless Mesh Network promises. However, business users require the same level of quality of services as for the 3G connections, and they become worried if the services are unreliable or unusable.

Thanks to the cheap installation of Wireless Mesh Networks, many applications can rely on that technology such as video surveillance. The owners of block of houses may use video surveillance in order to track illegal activities. Instead of installing cables, owners may choose a solution based on mesh routers that transmit the picture of cameras to the monitoring room through wireless hops. The owners may not want to employ full-time security personnel, but they want to have the
In order to take advantage of the fact that a Wireless Mesh Network is maintained by multiple operators or the mesh routers uses multiple channels, a new EU project was launched in FP7 called EU-MESH (http://www.eu-mesh.eu). Security related issues, that are introduced in Section 1.2.2, were addressed in this context. In this thesis, two of them are investigated: 1) fast authentication, and 2) misbehaving router detection. Solutions introduced here are also part of technical documents delivered in the project. The project started in 2008, and successfully ended in the third quarter of 2010.

As I have already described, business users require reliable network access independently of the technology while they are moving. The re-authentication of the users at new access points is essential, but the process could have intolerable delay. Especially when the access point where the mobile business user associated belongs to other operator than the previous access point. Novel fast authentication methods are proposed in Chapter 4. In particular, there were two different methods. The first one, based on HOKEY and IEEE 802.11r standards, while the second one is based on certificates.

For the certificate based fast authentication method, I proposed the weak key mechanism in Section 4.5.4, in order to extend the circle of potential mesh clients that can be authenticated within tolerable time using certificate based authentication. The weak key mechanism can be applied to any protocols where the delay of digital signatures is critical and only the authenticity of the messages must be assured, but not non-repudiation. As a particular example, this mechanism was applied when routing messages were flooded in the network using digital signatures. The performance of generating and verifying digital signatures was increased with the weak key mechanism.

The performance of the certificate based authentication method and the weak key mechanism was evaluated through real implementation. I used the context of hostapd and wpa_supplicant [Malinen, 2009] to embed my authentication messages to EAP format. Therefore, in IEEE 802.11 wireless networks, these mechanisms can be used with minor implementation improvements.

The behavior of mesh routers can be modified by an external attacker due to the lack of physical protection. Considering the video surveillance application, an attacker who gets control over a mesh router can achieve by falsifying the routing metrics that the wireless cameras forward the pictures through the misbehaving routers. An attacker can prevent delivering the pictures to the computer which should store the data by simply dropping all the messages. During the attack, any kind of illegal activities can be done without getting nabbed. Misbehavior of the mesh routers must be detected in order to provide reliable video surveillance.

A misbehaving router detection mechanism is also essential in multi-operator maintained Wireless Mesh Networks where an operator can change the behavior of its mesh routers in order to gain advantage over the competitors.

In Chapter 5, a misbehaving router detection mechanism for link-state routing protocols is proposed. The main advantage of the proposal is — in contrast to some current solutions — that it does not require the routers to keep their neighbors under observation during the message forwarding phase, which may have low performance in a multi-channel environment.

My mechanism has been implemented in the framework of OLSRd [olsrd, 2010], too. This implementation includes 1) counting the forwarded authentic messages, 2) sending the counter value to the destination gateway, 3) evaluating the counter values, 4) flooding the Node Trust Values in the network, and 5) taking into consideration the Node Trust Values when new routes are established.
In this thesis, the security of two instances of mobile ad-hoc networks are considered: Delay Tolerant Network and Wireless Mesh Networks.

A Delay Tolerant Network (DTN) is an infrastructureless network, where the message dissemination is performed by the participating mobile end-nodes in a store-carry-and-forward manner. In order to provide secure and reliable data forwarding, it is essential 1) to stimulate the cooperation among the nodes otherwise the data delivery ratio can be intolerably low, and 2) to prevent the traceability of nodes by enhancing the privacy in data forwarding.

In Chapter 2, in order to stimulate the cooperation in data dissemination, I propose barter as an exchange mechanism in Delay Tolerant Networks where messages are forwarded with a dissemination based approach. By means of simulations, I show that the proposed barter mechanism indeed encourages nodes to disseminate messages which results in faster delivery and higher delivery ratio.

I build a system model and run simulations in order to investigate the effects of selfishness in the considered Delay Tolerant Networks. The investigations are based on a metric called goodput, which reflects the delivery ratio and the speed of data delivery simultaneously. The goodput is calculated by means of simulations, thus, it is critical to consider its steady state value. I show analytically using a Markovian model that the goodput converges to a steady state value at an exponential rate. Therefore, the goodput values would not change considerably in longer simulations, whose length are determined empirically.

I propose barter as an exchange mechanism in this context resulting in a novel approach which does not require any central entity as payment schemes do, nor the observation of other participants as reputation schemes require. Using game theory, I show that in a wide range of parameters of the simulations, the Nash Equilibrium strategies dictate that the users collect and disseminate messages even if they are not interested in them. This means that the proposed barter approach indeed mitigates the disadvantageous effect of selfishness.

I show by means of simulations that the barter mechanism when the nodes follow the Nash Equilibrium strategy increases the delivery ratio and speeds up the delivery in those scenarios from the considered ones when the selfish behavior hinders the message dissemination. Furthermore, I show that in some scenarios, the goodput is close to the ideal case. All these are performed by comparing three different cases: 1) one with barter as the encouraging mechanism where the users follow one of the Nash Equilibrium strategies, 2) one when the nodes store only those messages which they are interested in and no encouraging mechanism is present, and finally 3) one when all the users forward all the messages without any restrictions.

In Chapter 3, for dissemination based Delay Tolerant Networks, I propose a method called Hide-and-lie in order to mitigate traceability of users. My proposal is beneficial against attacks where a user can be profiled based on the information on what messages the node stores and what messages it wants to download. Note that the users can be traceable even if each node communicates with the other nodes through anonymous links.
I adopt a system model built for the investigation of a barter mechanism, I build an attacker model, and I propose some specific attack algorithms for the considered Delay Tolerant Networks. I show by means of simulation that nodes are traceable in the considered model with high probability if no defense mechanism is applied. I investigate by analytical tools the limits of success probability of the attacks relying on the interest profile of the users.

I propose a general defense mechanism, called Hide-and-lie against the considered attackers. I show by means of simulations that the success probability of the attacker can be decreased almost to the level of simple guessing while the goodput of the nodes does not decrease considerably, furthermore, in some scenarios, the goodput increases.

The other considered mobile ad hoc network instance is the Wireless Mesh Network. A regular Wireless Mesh Network consists of mesh routers that form a static wireless ad hoc network as an infrastructure and mesh clients that use that infrastructure. As mesh networks are typically not stand alone networks, some of the mesh routers function as gateways. A subset of mesh routers function as wireless access points where mobile mesh clients can connect to the network.

I concentrate on Wireless Mesh Networks, where the infrastructure is maintained by multiple operators who provide broadband wireless access to the Internet for their customers based on contracts. Wireless mesh networks have to support user mobility and they have to fulfill QoS requirements, too, because mesh clients can move during the data transmission while they may run QoS aware applications.

For the secure and reliable data delivery in multi-operator maintained Wireless Mesh Networks, the following two issues have been addressed in this thesis: 1) In order to prevent unauthorized access to the network which may degrade the quality of services, it is essential to authenticate the mesh clients and agree on keys for access control enforcement when they associate to a new access point. The authentication should not last such long that it causes intolerable delay for QoS aware application. 2) It is also essential that the mesh routers detect and avoid in router selection the misbehaving routers.

In Chapter 4 firstly, I present a taxonomy for authentication and access control methods that have been proposed so far. I found that none of them can satisfy all the requirements on multi-operator maintained Wireless Mesh Networks. Therefore, I propose two different schemes: 1) one based on current standards (HOKEY and IEEE 802.11r), and 2) two certificate based authentication and access control enforcement protocols. In order to reduce the authentication delay, I propose a so called weak key mechanism for constraint devices.

I propose a certificate based authentication and access control enforcement protocol based on nonces and another one based on timestamps. I show by measurements on a real implementation that the authentication delay does not cause intolerable interruption during the handover for the QoS aware applications in case of powerful mesh clients and constraint access points which is the typical case. I show informally that my solutions fit to multi-operator maintained Wireless Mesh Networks.

I propose a variant of the proposed nonce and timestamp based protocols that allows constraint mesh clients to use shorter keys (weak keys). I show that a 30% reduction of authentication delay can be achieved by applying the weak key mechanism when constraint mesh clients are present. I show that the application of weak key mechanism is beneficial when the gain on the processing time of the public key cryptographic operations is larger than the loss on the longer certificate chain verification time.

In Chapter 5, I present a novel reputation system for detecting and avoiding misbehaving routers in Link-state Routing for Wireless Mesh Networks.

I propose a novel reputation system for detecting misbehaving routers in Link-state Routing for Wireless Mesh Networks. Each router’s reputation value is calculated over counters. Each router maintains a counter for each data flow and counts how many messages were forwarded in each flow. The counters are sent to the gateway that is at the end of the path. The gateway calculates a reputation value, called node trust value, for each router such that it counts on that a misbehaving router can send a fake counter value. I show by means of simulation that the proposed mechanism can differentiate misbehaving routers from honest ones.
I propose a mechanism with which the routers are considered in the path selection procedure with a probability that is proportional to their node trust value. I show by means of simulations that thanks to the detection and the route selection mechanism, the number of the dropped messages decreased around 50%, while the length of the routes increases only slightly.
Appendix

Authentication related standards

A.1 IEEE 802.11i

Today, IEEE 802.11i [IEEE Std 802.11i™, 2004], an amendment to the IEEE 802.11 standard [IEEE Std 802.11™(R2003), 2003] provides a security framework at the link between a mesh client and an access point. IEEE 802.11i has been incorporated in the IEEE 802.11-2007 standard [IEEE Std 802.11™-2007, 2007]. IEEE 802.11i provides various security services: access control, data confidentiality, data integrity and data origin authenticity.

After the STA and the AP make an association, they perform the authentication. IEEE 802.11i defines two methods for authentication: pre-shared key and centralized authentication. Here, I give an overview only about the centralized authentication because I consider operator maintained networks.

In the centralized authentication method, IEEE 802.11i utilizes the IEEE 802.1X framework for access control. The IEEE 802.1X setup is comprised of three entities: the supplicant (the term used by the standard for the user trying to gain access to the network), the authenticator and the authentication server (AS). In IEEE 802.11i, the mesh client plays the role of the supplicant and the access point plays the role of the authenticator. Standards IEEE 802.11i and IEEE 802.1X together provide a solution for key management: the so called Pairwise Master Key (PMK) derived at the end of a successful IEEE 802.1X authentication is utilized in the IEEE 802.11i key management phase to derive session keys. Figure A.1 shows the sequence of the centralized IEEE 802.11i authentication utilized in a standard IEEE 802.11i handoff process.

IEEE 802.1X utilizes a centralized authentication scheme. This means that the authentication is not done by the authenticator itself but by the AS. For this reason, the messages that are part of the IEEE 802.1X authentication are forwarded to the AS.

The method used for authentication is chosen by the AS. IEEE 802.1X does not make any restrictions, however, in practice exclusively the EAP framework is used with IEEE 802.11i. IEEE 802.1X does not define a mandatory protocol for the authenticator-to-AS communication either. However, typically the RADIUS protocol [Rigney et al., 2000] is used as a carrier protocol for EAP messages between the authenticator and the AS. Diamater, an upgrade to the RADIUS protocol has been proposed [Calhoun et al., 2003], but RADIUS is still considered to provide satisfactory authentication service.

It is the authenticator’s task to notify the AS of a new authentication request. The reply of the
A. AUTHENTICATION RELATED STANDARDS

AS initiates the EAP authentication and the authenticator forwards the authentication messages between the supplicant and the authentication server via EAPOL and RADIUS, respectively. The AS notifies the authenticator about the success or failure of the authentication. In the former case, the authentication server derives the PMK and sends it to the authenticator. The supplicant also derives the PMK. After a successful authentication, the authenticator initiates the key management phase. Handshake messages carried in EAPOL messages called EAPOL-Key frames are handled locally.

![IEEE 802.1X authentication](image)

Figure A.1: IEEE 802.1X authentication

The final step is the key management. After both the authenticator and the supplicant have obtained the PMK, they execute a handshake called four-way handshake. The four-way handshake serves three purposes:

- demonstrate the knowledge of the PMK,
- generate the connection keys, namely the Pairwise Transient Key (PTK) and Groupwise Transient Key (GTK),
- confirm cipher suite selection.

The sequence of the four-way handshake is shown in Figure A.2.

1. The authenticator generates the random number ANonce and sends it to the supplicant.
2. The supplicant generates the random number SNonce and generates the connection keys (PTK and GTK). It sends the authenticator a message containing SNonce, the GTK encrypted with the PTK, the initially chosen ciphersuites (SPA RSN IE) and a MIC (Message Integrity Code).
3. The authenticator also computes the PTK and sends a message containing Anonce, the initially advertised ciphersuites (AA RSN IE) and a MIC.

4. The authenticator sends a confirmation containing SNonce and a MIC.

Every message contains the identifier of the sender (SPA or AA), a sequence number (sn) and a message number identifier. These elements, along with the fresh nonces, all serve the goal of defending against man-in-the-middle and replay attacks, while the MIC provides integrity protection on 4-way handshake messages calculated with PTK.

Finally, if the handshakes are successful, the authenticator allows the supplicant to start data flow encrypting and protecting integrity with PTK.

In multi-operator environment. Basically, the standard IEEE 802.11i does not explicitly support the authentication at a multi-operator maintained Wi-Fi network, but the remote authentication server can do it and the RADIUS protocol does.

In Figure A.3, I show how the messages are exchanged between the participants when a mesh client authenticates to a foreign access point. When a mesh client, a subscriber of the operator $O_1$ associates to an access point belonging to operator $O_2$, the access point simply forwards all the messages to the dedicated authentication server belonging to operator $O_2$ (AS$_2$). The AS$_2$ recognizes that the mesh client belongs to different operator and it forwards the authentication messages to the authentication server belonging to operator $O_1$ (AS$_1$) with a proxy module. After the AS$_1$ authenticated the mesh client successfully, it sends an ‘Access-accept’ message with the MSK to the AS$_2$. The AS$_2$ forwards the ’Access-accept’ message to the access point containing the MSK key. For the access point, the proxying remains transparent.
A. AUTHENTICATION RELATED STANDARDS

A.2 IEEE 802.11r

In the IEEE 802.11-2007 [IEEE Std 802.11™-2007, 2007], eight amendments were merged including the IEEE 802.11i [IEEE Std 802.11i™, 2004] standard. The standard IEEE 802.11r [IEEE 802.11r™-2008, 2008]— also called Fast Transition (FT) — is a new amendment to IEEE 802.11-2007. This amendment is responsible for fast and secure handover in Wireless LAN networks to support QoS-aware applications. I give an overview of the IEEE 802.11r authentication in this section.

A.2.1 General description

When a mesh client first associates to an access point of the mesh network it performs a full authentication as it is done in IEEE 802.11i but extended with some IEEE 802.11r specific messages. The access point AP₀ through which this full authentication is performed will play a special role during the upcoming handoff processes. Before leaving the access point currently associated with, the mesh client indicates the handoff and the identity of AP₀ to the next access point (through the current access point or directly). The next access point obtains an authentication key K from AP₀. The mesh client is able to generate K using some public information and the initial authentication key shared with AP₀. The handoff is completed by running the 4-way handshake with the next access point and deriving connection keys from K.

In the standard IEEE 802.11r, some special roles are introduced. The AP₀ plays the role of R₀KH (Root₀ key holder) when generates access point specific key (PMK-R₁) from the MSK (Master Secret Key, the result of a full authentication), and all access points including AP₀ plays the role of R₁KH (Root₁ key holder) when obtains the PMK-R₁ and when derives the PTK. On the mesh client’s side, the same roles are present: S₀KH (Supplicant₀ key holder) generates the access point specific keys and S₁KH (Supplicant₁ key holder) derives the PTK with R₁KH.

A.2.2 Mobility domains

Each access point belongs to one mobility domain. The mobility domains are identified by a so called Mobility Domain Identifier (MDID). The MDID is advertised in beacon messages and in this way, the mesh clients can decide which access point to associate next. This may be important because the fast handover is assured only within the mobility domains as the inter-domain handover is not covered in the standard IEEE 802.11r.
A. AUTHENTICATION RELATED STANDARDS

A.2.3 Initial authentication

When a mesh client associates to a mesh network it performs a full, centralized initial authentication similar to a IEEE 802.11i authentication described in Section A.1. Some messages are extended with IEEE 802.11r specific elements. An overview of the exchanged messages can be seen in Figure A.4 where I emphasize the new elements.

Mesh client Access point Remote authentication center

Capability advertisement (FT, MD-ID)
802.11 Authentication request (Open)
802.11 Authentication response (Open)
(Re)association request (MD-ID)
(Re)association response (MD-ID, R0KH-ID, R1KH-ID)

802.1X EAP Authentication

ANonce
SNonce, MIC, PMKR1NAME, MD-ID
ANonce, MIC, PMKR1NAME, MD-ID, GTK, ReassocDeadline, KeyLifeTime
MIC

Access Accept (MSK)

The capability advertisement is extended with MDID and with an FT bit which indicates whether an access point supports the standard IEEE 802.11r. The mesh client should return this MDID in (Re)association request message and the access point checks if the value meets the advertised one. In the (Re)association response message, the access point sends again the MDID and adds its R0KH and R1KH identifier.

Then, the mesh client and the authentication server perform a IEEE 802.1X based authentication through the access point as it is defined in the standard IEEE 802.11i. In the end of the full authentication, the access point obtains the result if it is successfully completed or not. If it is successful, the access point obtains MSK from the authentication server and both the mesh client and access point calculate the access point specific PMK-R1.

Finally, the FT 4-way handshake is performed. This new 4-way handshake relies on the 4-way handshake introduced in the standard IEEE 802.11i, but these messages are also extended with some IEEE 802.11r specific data:

- The second message is extended with MDID and the name of the access point specific key (PMKR1NAME defined in next paragraph).
- The third message is extended with the aforementioned entries, but also contains the deadline for the next reassociation (ReassocDeadline) and the lifetime of the access point specific PMK-R1 (KeyLifeTime) entry. I explain both entries in the next paragraphs.

Figure A.4: Initial authentication in IEEE 802.11r
A. AUTHENTICATION RELATED STANDARDS

A.2.4 Key hierarchy

Recall that the access point specific PMK-R1 keys are derived from the MSK. This derivation is performed in two steps. First the R0KH generates a 256-bit long R0 key holder specific key (PMK-R0) and an identifier of the key (PMKR0NAME) which can be public. The key and its identifier is calculated over the MSK key, the MDID, an identifier of the R0KH which is unique in the mobility domain (R0KHID) and the MAC address of the mesh client (MACMC) as it is shown in Eqs. (A.1) and (A.2), respectively.

\[ PMK - R0 = \alpha_{MSK}(SSID, MDID, R0KHID, MACMC) \]  \hspace{1cm} (A.1)

\[ PMKR0NAME = \beta_{MSK}(SSID, MDID, R0KHID, MACMC) \]  \hspace{1cm} (A.2)

\( \alpha \) and \( \beta \) functions are based on HMAC-SHA256 function.

When needed, the R0KH calculates for any point in the same mobility domain an access point specific key (PMK-R1) and its identifier (PMKR1NAME) over the PMK-R0 and the MAC address of the mesh client and the access point (MACAP) as it is shown in Eqs. (A.3) and (A.4), respectively.

\[ PMK - R1 = \alpha_{PMK-R0}(MACAP, MACMC) \]  \hspace{1cm} (A.3)

\[ PMKR1NAME = \beta_{PMK-R0}(MACAP, MACMC) \]  \hspace{1cm} (A.4)

The PTK is derived from the PMK-R1, the MAC address of the mesh client and the access point and a random number per each participant generated during the FT 4-way handshake.

\[ PTK = \gamma_{PMK-R1}(SNonce, ANonce, MACAP, MACMC) \]  \hspace{1cm} (A.5)

\( \gamma \) function is also based on HMAC-SHA256 function.

As one key is generated from the other, a) the lifetime of the PTK shall not be longer than the lifetime of PMK-R1, b) the lifetime of the PMK-R1 shall not be longer than the lifetime of the PMK-R0, and c) the lifetime of the PMK-R0 shall not be longer than the lifetime of the MSK. If no lifetime is provided for the MSK, the lifetime of the PMK-R0 is maximized by the value \textit{KeyLifeTime} propagated in FT 4-way handshake. This value is equivalent in the whole mobility domain.

A.2.5 Re-authentication within a mobility domain

After an initial authentication, the mesh client can associate to another access point within mobility domain fast enough such that QoS-aware services will not interrupt. Here, the reassociation needs only four messages and the message exchanges can be performed in two parts. The last two messages are sent during the handover, while the first two messages can be exchanged before the mesh client disassociates from the current access point. This gives a great opportunity to the potential next access point to obtain the access point specific key before the handover from the R0KH. The first two messages can be sent through the current access point or over the air, directly. The two cases require two different frame formats, however I show both cases in Figure A.5 as the contents related to the reassociation process are the same.

The objective and the message contents of the Fast Transition Protocol are similar to the 4-way handshake in a sense that it proves the knowledge PMK (here PMK-R1) for both participants and they can derive the PTK. However, the Fast Transition Protocol has some other objectives which cause some differences:

- The access point is not aware who is the R0KH of the mesh client. This information is sent in the first message with PMKR0NAME which identifies the related PMK-R0 key in the database of R0KH for the PMK-R1 calculation.

- The second message does not contain Message Integrity Code (MIC), thus the access point does not have to obtain the PMK-R1 from the R0KH until the second message.
A. AUTHENTICATION RELATED STANDARDS

Figure A.5: Reassociation in IEEE 802.11r (Fast Transition Protocol)

- R1KH-ID is sent in the second message which information is needed to the mesh client for the calculation of PMK-R1.
- In the first two messages, the PMKR0NAME is added, but after the mesh client is able to calculate PMK-R1, the PMKR1NAME is sent which implies the knowledge of PMKR0NAME.
- Last two messages sent during the handover contains the following information: MDID, R0KH-ID and R1KH-ID. These entries counts when the MIC is calculated and the mesh client and the access point prove to each other that they sent these information in the first two messages, too.

If the first two messages are sent to the target through the current access point, the mesh client has to declare the target access point. The current access point, then, forwards between the mesh client and the target access point encapsulating and decapsulating the messages into a so called remote request protocol specified in the standard.

The standard IEEE 802.11r defines a protocol not detailed here in which a mesh client can reserve resources in the access point. This information and the list of successfully reserved resources is exchanged in two further messages before the mesh client disassociates from the current access point.

The standard IEEE 802.11 studies only the medium access control and the physical layer, while the process of the request and the distribution of the PMK-R1 is easiest to implement in the network layer. Therefore, these processes are not standardized in the standard IEEE 802.11r, not even stated how and when to request and distribute the PMK-R1s. My description relies on [Clancy, 2008] whose author is one of the author of the standard IEEE 802.11r.

A.3 HOKEY

The Handover Keying (HOKEY) is an extension to the EAP standard [Aboba et al., 2004] and consists of five RFC standards (RFC 5169 [Clancy et al., 2008], RFC 5295 [Salowey et al., 2008],...
A. AUTHENTICATION RELATED STANDARDS

RFC 5296 [Narayanan and Dondeti, 2008], RFC 5749 [Hoeper et al., 2010], and RFC 5836 [Ohba et al., 2010]). As the standard HOKEY extends the EAP standard, it may suit wide variety of networks or applications, however, I consider it in the context of multi-operator maintained mesh networks.

A.3.1 General description

In the HOKEY standard, the authors consider a similar scenario than in the IEEE 802.11i. However, instead of performing full authentication with a remote authentication server each time when a mesh client re-authenticates to the mesh network at different access points, the HOKEY supports reusing the key generated at the initial authentication and shorten the delay of the re-authentication process. It is reached through the following steps:

- During the initial authentication process, the client and the authentication server generates a handover integrity key.
- When the client authenticates to another access point, it sends a request extending with a Message Authentication Code (MAC) using the handover integrity key in a special EAP format (EAP-Initiate/Re-auth Packet) defined in RFC 5296 [Narayanan and Dondeti, 2008].
- The authentication server, after verifying the MAC, responds to the access point with a fresh key generated in message EAP-Finish/Re-auth Packet such that the mesh client is able to generate it, too, without any further communication.

The main benefit of this solution is that only one round trip message exchange is required between the client and the remote authentication server to perform the re-authentication. HOKEY was designed such that the operations performed during the re-authentication is independent of the initial EAP algorithm and the way of generating the first key. Furthermore, the HOKEY can handle different domains and it supports to install local authentication servers to shorten the round trip between the access point and the authentication server.

A.3.2 Key hierarchy

A successful EAP authentication results in two shared keys between two participants: Master Secret Key (MSK) and Extended MSK (EMSK). In IEEE 802.11i, only the MSK is used as a PMK and it leaves the EMSK out of the consideration. In the HOKEY, the EMSK is used for deriving keys for the re-authentication in such a way that each domain obtains independent key. Because of the domains, the key generation is hierarchical as it is shown in Figure A.6.

![Figure A.6: Key hierarchy in HOKEY](image_url)
A. AUTHENTICATION RELATED STANDARDS

In RFC 5295, the HOKEY standard names the function, which derives one key from the other, Key Derivation Function (KDF). The participants can choose what function to use as a KDF, but in the standard there is a default specification, too.

From the EMSK, Domain-Specific Root Keys (DSRK) can be derived in the following way:

\[
DSRK = KDF(EMSK, "dsrk@ietf.org" \| \"domain\_name\|optional\_data\|length\) \quad (A.6)
\]

The DSRKs are sent to each domain when requested or in a proactive way defined in [Hoepner et al., 2010]. For the derivation of further keys from DSRK, the authentication server of the domain is responsible.

The re-authentication Root Key (rRK) is derived from EMSK or DSRK when general or domain specific keys are desired, respectively. The participants are allowed to derive other application specific keys beyond the rRK, but this is out of the scope of the HOKEY standard and my focus. The rRK is generated to derive other keys needed in re-authentication mechanism. The rRK is derived in the following way:

\[
rRK = KDF(EMSK \text{ or } DSRK, \text{"EAP Re-authentication Root Key@ietf.org" \| \"0\|length\) \quad (A.7)
\]

The re-authentication Integrity Key (rIK) is one of the keys required to perform fast re-authentication. As I have already stated in the general description, the rIK is used to authorize the authentication server to deliver unique keys to authenticators. The authenticator is an entity where the mesh client authenticates itself, typically access points are authenticators in mesh networks. The rIK is derived from rRK in the following way:

\[
rIK = KDF(rRK, \text{"Re-authentication Integrity Key@ietf.org" \| \"0\|cryptosuite\|length\} \quad (A.8)
\]

The cryptosuite defines the algorithm of MAC in messages EAP-Initiate/Re-auth Packet and EAP-Finish/Re-auth Packet. Now, the HOKEY standard supports HMAC-SHA MAC algorithms with different length of output.

The per-authenticator key (rMSK), which substitutes the MSK, is derived in the following way:

\[
rMSK = KDF(rRK, \text{"Re-authentication Master Session Key@ietf.org" \| \"0\|SEQ\|length\}) \quad (A.9)
\]

The uniqueness of the per-authenticator keys is provided by SEQ which variable is increased by one each time a new key is derived from rRK.

A.3.3 Initial authentication

During the HOKEY initial authentication, as it is shown in Figure A.7, the mesh client follows the base EAP standard and the authenticator receives the MSK as it is defined in RFC 3748 [Aboba et al., 2004]. While, the foreign authentication server (AS) extends the AAA (e.g. RADIUS [Rigney et al., 2000] or DIAMETER [Calhoun et al., 2003]) packets with a DSRK request message and a domain identifier which is used to generate DSRK. After a successful EAP authentication, the Home AS derives the DSRK from the EMSK and extends the EAP Success message with it. Note that after the initial authentication, the authenticator obtains the MSK instead of rMSK because of backward compatibility reason.
A. AUTHENTICATION RELATED STANDARDS

A.3.4 Re-authentication

The rIK is used and rMSK is derived when a HOKEY re-authentication is performed. A sample authentication method is shown in Figure A.8. The authenticator can indicate that it supports the HOKEY fast handover mechanism with an “EAP Initiate/Re-auth Start” message. The mesh client first sends an “EAP Initiate/Re-auth” message which is forwarded by the authenticator to the dedicated AS (Foreign AS) embedded into AAA protocol message. This message includes the SEQ number and a MAC calculated over the whole message. SEQ number is needed to the Foreign AS which calculates the next rMSK for the authenticator as it is shown in Eq. (A.9). The Foreign AS sends back an “EAP Finish/Re-auth” message which contains the required rMSK. The authenticator reads the rMSK which is sent encrypted with shared key between the Foreign AS and the authenticator. The HOKEY re-authentication protocol is finished by forwarding the EAP message to the mesh client.

A.3.5 Key delivery

In [Hooper et al., 2010], the delivery of keys belonging to different levels of the key hierarchy can be performed in a proactive way, e.g. a Home AS can deliver DSRK key to a Foreign AS before it requests. The mechanism uses RADIUS packet form.
A. AUTHENTICATION RELATED STANDARDS

Figure A.8: **HOKEY re-authentication in a foreign network**
Appendix B

Time consumption of asymmetric cryptographic primitives

I measured the time consumption of some widely known and analyzed public key based key exchange, digital signature and encryption algorithms listed below.

- **Key exchange algorithms**
  - Diffie-Hellman [Diffie and Hellman, 1976]
  - Elliptic Curve Diffie-Hellman [Certicom Research, 2000]

- **Digital signature algorithms**
  - RSA [Rivest et al., 1978]
  - DSA [FIPS PUBS, 1994]
  - Elliptic Curve DSA (ECDSA) [Certicom Research, 2000]

- **Cipher algorithms**
  - RSA
  - Elliptic Curve ElGamal (ECELG) [Rabah, 2005]

These cryptographic primitives are already implemented in various crypto libraries. For my measurements, I chose the Open SSL library [OpenSSL, 2010], because of the following reasons: 1) it is a widely used open source library, 2) each crypto primitive is already implemented in it (except for ECELG, but general operations over different elliptic curves are supported), 3) it is available in a crosscompiled version for each architecture which is supported by the OpenWRT [OpenWRT, 2010] embedded Linux distribution.

I measured the time consumption of the above listed algorithms with different key sizes or using different elliptic curves. In each case, I considered the average value of 100 measurements.

In the case of Diffie-Hellman key agreement algorithms, I measure the time needed to compute the common key by the two protocol participants. In the case of digital signature algorithms, I measure the time of generating and verifying signatures on a single block of data. The measurements of the encryption algorithms are performed in the same way as in the case of digital signature: I measure the time consumption of the encryption and the decryption, of a single randomly generated data block.

Herein, I define what key parameters were considered in different public key crypto algorithms. In the case of non-EC algorithms, I measured the time consumption with 256, 384, 512, 1024 and 2048 bit long keys. The generator number of DSA was always 5 and the prime was generated
randomly in each run. The exponent of the public key of the RSA algorithm is 65537 both in the case of encryption and digital signature. The elliptic curve based algorithms require to define an elliptic curve on which the operations can be performed. The OpenSSL implements the elliptic curves proposed and standardized in three different documents issued by three different organizations: SECG [Standards for Efficient Cryptography Group (SECG), 2000], ANSI X9.62 [Accredited Standards Committee X9, 2005], and WAP [Wireless Application Forum, 1999]. In Table B.1, I describe the properties of elliptic curves I considered while measuring the time consumption.

ECRYPT II [FP7 ECRYPT II, 2007] recommendation for 2008 says that the 1024 bit asymmetric keylength and elliptic curves with 160 bit are sufficient for short-term protection (some years). And 816 bit asymmetric keylength and elliptic curves with 128 bit are not sufficient for confidentiality, and offer only very short-term protection for other purposes.

I have measured the time consumption of crypto primitives in two different devices. One device is a regular access point which will be responsible for authenticating the mesh clients in my proposed mechanism. The access point is a Linksys wireless router (WRT54GL v1.1) with 200 MHz MIPS CPU, 16 MB RAM and 4 MB Flash. The other one is a more powerful device as the mesh clients have usually more power than an access point, I used a desktop PC with Core2Duo 6400 CPU and 1 GB RAM.

In Figure B.1, B.2, and B.3, I plotted the results of the measurement of key exchange, digital signature and encryption algorithms, respectively. On the $x$ axis, I show the delay of the firstly measured operation (i.e., calculation of the common key by the first party, generation of a digital signature, and encryption), while on the $y$ axis, I show the time consumption of the secondly measured operation (i.e., calculation of the common key by the second party, digital signature verification, and decryption). The placing of a point, which is related to a specific algorithm with a specific key parameter, shows the delay of the two operations. The name of the algorithms and the key parameters are indicated in the legend of the figures. Elliptic curves correspond to the abbreviations in Table B.1; the standards and the OpenSSL library use the same or similar names.

For the sake of better readability, I grouped the points into groups and the points are indicated with the same mark in Figure B.1, B.2, and B.3 if the points are close to each other.

The DH and ECDH key exchange algorithms are for establishing shared secret between two parties which has no prior knowledge of each other, but the algorithms themselves do not provide

### Table B.1: Test elliptic curve parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Organization</th>
<th>Size</th>
<th>RSA/DSA</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>secp112r1</td>
<td>SECG</td>
<td>112</td>
<td>512</td>
<td>$\mathbb{F}_p$</td>
</tr>
<tr>
<td>secp112r2</td>
<td>SECG</td>
<td>112</td>
<td>512</td>
<td>$\mathbb{F}_p$</td>
</tr>
<tr>
<td>secp128r1</td>
<td>SECG</td>
<td>128</td>
<td>704</td>
<td>$\mathbb{F}_p$</td>
</tr>
<tr>
<td>secp128r2</td>
<td>SECG</td>
<td>128</td>
<td>704</td>
<td>$\mathbb{F}_p$</td>
</tr>
<tr>
<td>secp160r1</td>
<td>SECG</td>
<td>160</td>
<td>1024</td>
<td>$\mathbb{F}_p$</td>
</tr>
<tr>
<td>secp160r2</td>
<td>SECG</td>
<td>160</td>
<td>1024</td>
<td>$\mathbb{F}_p$</td>
</tr>
<tr>
<td>sect113r1</td>
<td>SECG</td>
<td>113</td>
<td>512</td>
<td>$\mathbb{F}_{2^m}$</td>
</tr>
<tr>
<td>sect113r2</td>
<td>SECG</td>
<td>113</td>
<td>512</td>
<td>$\mathbb{F}_{2^m}$</td>
</tr>
<tr>
<td>sect131r1</td>
<td>SECG</td>
<td>131</td>
<td>704</td>
<td>$\mathbb{F}_{2^m}$</td>
</tr>
<tr>
<td>sect131r2</td>
<td>SECG</td>
<td>131</td>
<td>704</td>
<td>$\mathbb{F}_{2^m}$</td>
</tr>
<tr>
<td>sect163k1</td>
<td>SECG</td>
<td>163</td>
<td>1024</td>
<td>$\mathbb{F}_{2^m}$</td>
</tr>
<tr>
<td>sect163r2</td>
<td>SECG</td>
<td>163</td>
<td>1024</td>
<td>$\mathbb{F}_{2^m}$</td>
</tr>
<tr>
<td>c2pnb163v1</td>
<td>X9.62</td>
<td>163</td>
<td>1024</td>
<td>$\mathbb{F}_{2^m}$</td>
</tr>
<tr>
<td>c2pnb163v2</td>
<td>X9.62</td>
<td>163</td>
<td>1024</td>
<td>$\mathbb{F}_{2^m}$</td>
</tr>
<tr>
<td>wtls1</td>
<td>WAP</td>
<td>113</td>
<td>512</td>
<td>$\mathbb{F}_{2^m}$</td>
</tr>
<tr>
<td>wtls5</td>
<td>WAP</td>
<td>163</td>
<td>1024</td>
<td>$\mathbb{F}_{2^m}$</td>
</tr>
<tr>
<td>wtls8</td>
<td>WAP</td>
<td>112</td>
<td>512</td>
<td>$\mathbb{F}_p$</td>
</tr>
<tr>
<td>wtls9</td>
<td>WAP</td>
<td>160</td>
<td>1024</td>
<td>$\mathbb{F}_p$</td>
</tr>
</tbody>
</table>
B. TIME CONSUMPTION OF ASYMMETRIC CRYPTOGRAPHIC PRIMITIVES

Figure B.1: Time consumption of key exchange algorithms

authenticity. Two things can be drawn from the previous statement: 1) Assuming that the connection key is derived from the result of the key exchange algorithm, the key size must be secure enough otherwise an attacker who logged the communication can decrypt it after breaking the key; 2) Beside the key exchange algorithm some other public key algorithms (typically digital signature) should be used to provide authenticity.

In Figure B.1(a) and B.1(b), one can see the time consumption of the key exchange algorithms performed in AP and PC, respectively. As it is expected, the delay of calculating the common key is independent of which private key is used when the common key is calculated.

Considering the time consumption of DH algorithm in AP, I can derive that this mechanism is not promising nowadays, as the delay is much longer for secure key sizes (1024 bit – 300 ms) than a QoS-aware application can tolerate. This is the case for the ECDH in most cases, however some elliptic curves (c2pnb163v2, sect163k1, wtls5 and wtls9) with secure key sizes can be calculated around 50 ms in my representative AP. In any other considered cases, the ECDH algorithm is either slow or insecure.

As Figure B.1(b) shows, a PC is powerful enough to perform such algorithms. Unfortunately, it does not accelerate the key agreement process as the access point and the mesh client has to perform the generation of the common key in parallel.

Note that in Figure B.1(a) and B.1(b), I do not indicate the time consumption of the DH with 2048 bit key size, because it has such a long delay that it would decrease the readability of the figures. In a powerful PC, it needs around 37 ms and in a less powerful AP, it lasts around 1.79 s.

In Figure B.2(a) and B.2(b), the time consumption of the digital signature can be seen in the case of AP and PC, respectively. The three analyzed algorithms show two different properties with the predefined key parameters. The RSA algorithm with small public key exponent is two or three degree quicker when the digital signature is verified compared to the process of signing. The ECDSA and the DSA are the opposite of RSA, the generation is two or three degree quicker than the verification if the operations are performed in the same powerful device. The reason is that in the case of signature generation the most time consuming operations can be performed without the knowledge of the data that has to be signed.

Note that one operation is time consuming considering the analyzed algorithms, and the other one only needs 1-2 ms even if the device has limited capacity. Furthermore, in the case of a powerful device (like PC), only the signature generation of RSA with 2048 bit needs considerable time. In the case of less powerful device (AP), from Figure B.2(a), one can read that an AP is not able to perform the operation which has higher delay (generation of digital signature in the case of RSA and verification in the case of DSA and ECDSA) with secure key sizes within a considerable time.
B. TIME CONSUMPTION OF ASYMMETRIC CRYPTOGRAPHIC PRIMITIVES

In Figure B.3(a) and B.3(b), the time consumption of the encryption algorithms can be seen in the case of AP and PC, respectively. Again, I can observe that the public key operation of the RSA, i.e. herein the encryption, is two or three orders of magnitude quicker than the decryption and does not cause more than 1-2 ms delay either in a limited device. The decryption process of the RSA with secure key sizes in AP needs more time than a QoS-aware application can tolerate. Considering the ECELG algorithm, the decryption is quicker than the encryption, but approximately two times, only. While, the quickest decryption with sufficiently large key size lasts more than 50 ms on average. Thus, the decryption can not be performed in limited device in any case during the handover. A more powerful device (PC) is able to encrypt and decrypt within the considered time with any kind of considered algorithms and with any kind of considered key parameter, except for the RSA decryption with 2048 bits (lasts 15 ms on average).

The conclusion is that using asymmetric key crypto in a powerful device does not preclude the possibility of seamless handover. While on a limited device, such as an AP, some crypto primitives cause too long delays. Here, I collect those from the set of considered algorithms and key parameters which can be performed with a sufficiently short delay:

- Digital signature generation with any key size using DSA or ECDSA
B. TIME CONSUMPTION OF ASYMMETRIC CRYPTOGRAPHIC PRIMITIVES

- Digital signature verification with any key size using RSA
- Any digital signature operation with a weak key
- Encryption with any key size using RSA

Thus, if the mesh client is as constraint as the considered access point, then they are not able to generate a shared secret securely within a considerable time. Furthermore, they are not able to generate and verify digital signatures using securely large keys. The main consequence is that two computationally constraint devices are not able to authenticate each other and to perform a DH key agreement protocol within the time that a seamless handover requires.
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>Access point</td>
</tr>
<tr>
<td>ARM</td>
<td>Advanced RISC Machine or Acorn RISC Machine</td>
</tr>
<tr>
<td>BIONETS</td>
<td>BIOlogically inspired NETwork and Services</td>
</tr>
<tr>
<td>BWM</td>
<td>Blake-Wilson and Menezes Provably Secure Key Transport Protocol</td>
</tr>
<tr>
<td>CAPWAP</td>
<td>Control And Provisioning of Wireless Access Points</td>
</tr>
<tr>
<td>CAT</td>
<td>Category identifier</td>
</tr>
<tr>
<td>CCTV</td>
<td>Closed-circuit television</td>
</tr>
<tr>
<td>CA</td>
<td>Certificate Authority</td>
</tr>
<tr>
<td>CRL</td>
<td>Certificate Revocation List</td>
</tr>
<tr>
<td>DoS</td>
<td>Denial-of-Service</td>
</tr>
<tr>
<td>DTN</td>
<td>Delay Tolerant Network</td>
</tr>
<tr>
<td>EAP</td>
<td>Extensible Authentication Protocol</td>
</tr>
<tr>
<td>EAPOL</td>
<td>EAP over LAN</td>
</tr>
<tr>
<td>EU</td>
<td>European Union</td>
</tr>
<tr>
<td>FP</td>
<td>Framework Programme</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile Communications</td>
</tr>
<tr>
<td>GW</td>
<td>Gateway</td>
</tr>
<tr>
<td>HLS</td>
<td>Hide-and-Lie Strategy</td>
</tr>
<tr>
<td>HOKEY</td>
<td>Handover Keying</td>
</tr>
<tr>
<td>IAPP</td>
<td>Inter Access Point Protocol</td>
</tr>
<tr>
<td>ID</td>
<td>Identity</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IP</td>
<td>Interest Profile</td>
</tr>
<tr>
<td>IPsec</td>
<td>Internet Protocol Security</td>
</tr>
<tr>
<td>ISP</td>
<td>Internet Service Provider</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LAS</td>
<td>Local Authentication Server</td>
</tr>
<tr>
<td>MAC</td>
<td>Message Authentication Code</td>
</tr>
<tr>
<td>MAC address</td>
<td>Media Access Control address</td>
</tr>
<tr>
<td>MANET</td>
<td>Mobile Ad Hoc Network</td>
</tr>
<tr>
<td>MC</td>
<td>Mesh client</td>
</tr>
<tr>
<td>MR</td>
<td>Mesh router</td>
</tr>
<tr>
<td>MSK</td>
<td>Master Session Key</td>
</tr>
<tr>
<td>Multi-WMN</td>
<td>Multi-operator maintained WMN using multiple channels</td>
</tr>
<tr>
<td>NTV</td>
<td>Node Trust Value</td>
</tr>
<tr>
<td>OP</td>
<td>Operator</td>
</tr>
<tr>
<td>PANA</td>
<td>Protocol for carrying Authentication for Network Access</td>
</tr>
<tr>
<td>ACRONYM</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>PMK</td>
<td>Pairwise Master Key</td>
</tr>
<tr>
<td>PTK</td>
<td>Pairwise Transient Key</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Services</td>
</tr>
<tr>
<td>RADIUS</td>
<td>Remote Authentication Dial In User Service</td>
</tr>
<tr>
<td>RAS</td>
<td>Remote Authentication Server</td>
</tr>
<tr>
<td>RRW</td>
<td>Restricted Random Waypoint mobility model</td>
</tr>
<tr>
<td>RW</td>
<td>Random Walk mobility model</td>
</tr>
<tr>
<td>SEQ</td>
<td>Sequence number</td>
</tr>
<tr>
<td>SUMO</td>
<td>Simulation of Urban MObility</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
<tr>
<td>UP</td>
<td>User Profile</td>
</tr>
<tr>
<td>VANET</td>
<td>Vehicular Ad-hoc Network</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice over IP</td>
</tr>
<tr>
<td>WMN</td>
<td>Wireless Mesh Network</td>
</tr>
</tbody>
</table>
List of publications


LIST OF PUBLICATIONS


Bibliography


